

**Т. П. Дяк, Ю. І. Грицюк***Національний університет "Львівська політехніка", м. Львів, Україна*

ПРОЕКТУВАННЯ СИСТЕМИ АВТОМАТИЗОВАНОГО ГЕНЕРУВАННЯ ВІРШОВАНИХ ТВОРІВ

Розглянуто особливості проектування системи автоматизованого генерування віршованих творів, що відкриває нові можливості художнього мовлення та сфери шоу-бізнесу, насамперед підготовки віршів і пісень. Доволі часто тексти пісень без особливого змісту стають успішними через відсутність складних сюжетів, а також через ненав'язливість і легкість їхнього сприйняття слухачами. Проаналізовано відомі літературні джерела та наявні програмні продукти, які можуть генерувати віршовані твори, поєднуючи різні методи та алгоритми. Встановлено, що жоден з них не здатен забезпечити змістовність і унікальність віршованого твору водночас, тим більше українською мовою. Проаналізовано наявні підходи до генерування віршованих творів, серед яких актуальними є метод на підставі шаблонів, генерування та тестування, еволюційні алгоритми та метод на підставі конкретних випадків. Досліджено особливості генерування віршованих творів, насамперед правила римуння, види строф, віршовані ритми та розміри. Розроблено підхід до автоматизованого генерування віршованих творів з використанням еволюційних алгоритмів і методу на підставі конкретних випадків. Їхнє поєднання нагадує послідовність дій для творчих особистостей під час створення віршів або написання текстів пісень. Розглянуто особливості організації нейронної мережі для автоматизованого генерування віршованих творів. Запропоновано навчання нейронної мережі виконати за методом зворотного поширення та з використанням генетичного алгоритму. Проаналізовано принцип роботи алгоритмів пошуку оптимальних рішень, які містять такі послідовні етапи як ініціалізацію, оцінювання рішень, відбір популяцій, еволюцію рішень. Детально досліджено їхню взаємодію та різні можливості для навчання нейронної мережі. Розроблено алгоритм, за яким програмний додаток буде аналізувати запропоновані користувачем віршовані твори та генерувати нові його варіанти на підставі отриманих від нейронної мережі логічно зв'язаних слів чи рядків куплета вірша. Користувач може вносити правки як до складових вірша, так і до згенерованих віршованих творів, і в такий спосіб може навчати нейронну мережу. Розроблено специфікацію вимог до програмного додатку, визначено основні вимоги до користувацького інтерфейсу, а також встановлено потенційні класи користувачів, які будуть його використовувати.

Ключові слова: комп'ютерна лінгвістика; штучний інтелект; нейронна мережа; генетичний алгоритм; оптимальне рішення.

Вступ

Сьогодні інформаційні технології вже не тільки замінили механічну роботу людини, а й починають вирішувати більш творчі завдання, такі як створення музичальних композицій, компонування відео, тощо [1], [10], [26], [37]. Написання поезії належить до тих творчих завдань, куди штучний інтелект також починає заходити, маючи для цього всі технічні можливості, щоб опанувати і цей вид людської діяльності [4], [20], [36], [44]. На даний момент вже існують програмні засоби для пошуку рими, визначення віршованого розміру, що допомагають творчим особистостям у підготовці віршів і пісень [15], [16], [17], [18], [38], [39]. Написання текстів пісень займає багато часу та вимагає творчої роботи як поетів і композиторів, так і співаків [9], [18], [41], [45].

Сучасні програмні додатки для генерування віршованих творів відкривають нові можливості для сфери шоу-бізнесу, наприклад, підготовки пісень. Доволі часто тексти пісень без особливого змісту стають успішними через відсутність складних сюжетів, а також ненав'язливість і легкість їхнього сприйняття слухачами [16], [23], [37]. Саме такі тексти пісень можуть генерувати програми, використовуючи для цього еволюційні алгоритми [20], [36]. Вони автоматизують процес підготовки віршованих творів, інколи навіть із логічним змістом [17], [18], [23], [51], [52]. Звісно, сучасні програми ще не можуть повністю генерувати віршовані твори певного змісту хоча б так, як це робить поет-початківець, а про справжнього митця – годі й мріяти. Тому основне завдання таких програм полягає в побудові логіч-

но зв'язаних словосполучень, котрі будуть дотримуватись приблизно одного стилю та форми [15], [50]. Загалом це дасть змогу потенційному поету чи піснярю деяке підґрунтя для творчих роздумів, бажання щось творити і початку їхнього натхнення.

Однією з проблем вирішення такого творчого завдання є розроблення адекватного алгоритму, який би на підставі проаналізованих текстів віршів чи пісень і аналізу їхнього змісту генерував нові віршовані твори [2], [5], [18], [44], [51]. Не секрет, що за аналогією працює і підсвідомість людини: насправді те, що людина вважає виявом своєї творчості, є нічим іншим, як синтезом попередньо прочитаного, почутого чи побаченого [2], [18]. Адже "все нове – це добре забуте старе", як висловився французький письменник Жак Пеше ще у 1824 році. Окрім цього, розроблений відповідний додаток дасть змогу творчим особистостям значно полегшити підготовку віршів і пісень, порівнювати отримані результати його роботи із власними спробами, оцінювати його практичну значущість та визначати шляхи покращення отриманих результатів.

Об'єкт дослідження – автоматизація генерування віршованих творів.

Предмет дослідження – методи і засоби побудови логічних зв'язків між словами та генерування з них римованих текстів, придатних для автоматизованого генерування віршів і пісень, правильних за формою подання та логічних за змістом.

Мета роботи – генерування віршованих творів на підставі логічно зв'язаних структур слів (речень, строф, рядків), які матимуть заданий зміст і будуть пов'язані з іншими структурами твору.

Для досягнення зазначеної мети необхідно вирішити такі *основні завдання дослідження*:

- проаналізувати проблему автоматизованого генерування віршованих творів, розглянути наявні методи їх підготовки та різні програмні додатки, що реалізують ці методи;
- дослідити сучасні підходи та методи для вирішення завдання автоматизованого генерування віршованих творів, скласти алгоритм побудови логічних зв'язків між словами, розробити специфікацію вимог до програмного додатку, а також вимоги до користувацьких інтерфейсів.

Наукова новизна отриманих результатів дослідження – розроблено метод генерування логічно зв'язаних текстів і підготовка римованих клаузул віршованих творів на їхній підставі; розроблено підхід до генерування віршованих творів, який поєднує еволюційний алгоритм і метод на підставі окремих випадків, що дає змогу творчій особистості задавати ключові слова, тематику віршованого твору або певну його частину.

Практична значущість результатів дослідження – розроблений програмний додаток "GeneratorPC" дасть змогу генерувати початкові рядки і куплети віршованого твору за попередньо заданими параметрами та на підставі запропонованих текстів, а також веб-систему, яка відображає отримані результати безпосередньому її користувачу.

Аналіз останніх досліджень та публікацій. Серед вітчизняних науковців заслуговує уваги робота [41], в якій здійснено огляд науково-лінгвістичних підходів, систематизовано дослідження вітчизняних і закордонних вчених щодо семантичних і функціональних особливостей ритміко-синтаксичної організації текстів су-

часної американської поезії, окреслено перспективи подальших лінгвістичних досліджень. Для розкриття теми дослідження автором проведено огляд наукового доробку, визначено підходи різних шкіл і течій щодо проблеми ритміко-синтаксичної організації віршованих творів, до якої зверталися на початку XIX століття поети-символісти (А. Білий, В. Я. Брюсов, С. С. Бобров), представники формальної поезики (Б. В. Томашевський [45], Ю. М. Тинянов [46], В. М. Жирмунський, О. М. Пешковський, В. Б. Шкловський, В. Б. Холшевников, Р. О. Якобсон [22], О. М. Брік [3], Б. І. Ярхо, Б. М. Ейхенбаум [8] тощо), структурно-семіотичної поезики (Ф. де Сосюр, А. Сеше, Ю. М. Лотман, Р. О. Якобсон [22], М. О. Краснопорова, А. М. Колмагоров [25], К. Ф. Тарановський, Б. М. Ейхенбаум [8], Г. О. Вінокур, Ю. М. Тинянов [46], [47], О. Н. Грінбаум, І. Р. Гальперін), лінгвопоезики (Г. М. Поспелов, М. Г. Тарлінська, Н. П. Неборсіна, Т. В. Скулачова, М. Л. Гаспаров [9], [10], [11], Л. І. Белехова [1]).

Однак, проблему автоматизованого генерування віршованих творів почали досліджувати закордонні вчені, а саме: Cichocki A. і Unbehauen R. [5], Du K.-L. і Swamy M. N. S. [7], Geqay R. і Liu T. [12], Gervas P. [13], Girshick R. і Donahue J. [14], Hugo Gonçalo Oliveira і Alberto Díaz [16], Jie Wang і Chengzhi Zhang [23], Kamenka Staykova [44], Mandic D. P. і Chambers J. A. [27], Manurung H. [28], Mazur M. [29], Oliviera H. G. [35], Ren S. і He, K. [40], Yuanzhi Ke і Masafumi Hagiwara [51]. Проаналізуємо їхні розробки дещо детальніше.

У роботі [44] автор описує особливості генерування природної мови NLG (англ. *Natural Language Generation*) та семантичні технології, які можуть бути використані для цього. Автор стверджує, що проблема генерування природних мов належить до двох областей – комп'ютерної лінгвістики та штучного інтелекту. Практично всі сучасні дослідження NLG в основному зосереджені на створенні комп'ютерних систем або додатків, які генерують зрозумілі тексти природними мовами. Розроблені системи NLG, зазвичай, беруть на вхід деякі немовні інформаційні подання, використовують знання про природну мову та знання про певну область для створення відповідних документів, звітів, пояснень, резюме та інших типів текстів. У роботі проаналізовано розробки ІТ-галузі для генерування природної мови з її моделями, техніками, застосуваннями, а також результати дослідження стосовно того, як семантичні технології поступово втягуються у вирішення проблеми генерування тексту.

У роботі [18] автор описує особливості автоматичного генерування текстів пісень у семантичному домені, і реалізованому в програмній системі Tra-la-Lyrics 2.0. У попередній версії цієї системи основна увага була спрямована на створення тексту, де наголоси відповідали ритму заданих мелодій. Там не було побоювання щодо того, чи має згенерований текст сенс, чи виділеним словам спільна семантична асоціація. У наявній версії системи тексти пісень можна генерувати в семантичному домені, визначеному одним або декількома початковими словами. Також автор передбачив інтеграцію оригінального ритм-модуля Tra-la-Lyrics в систему PoeTryMe [39], тобто загальну платформу, яка породжує поезію із семантично зв'язними реченнями. Щоб встановити свій прогрес, ритм, рими та семантичну уз-

годженість у текстах, створених оригінальною системою Tra-la-Lyrics, були проаналізовані та порівняні з текстами, створеними за допомогою нової версії цієї системи Tra-la-Lyrics 2.0. Аналіз згенерованих текстів показав, що у ліриці за новою системою слова мають вищу семантичну асоціацію між ними і з заданим змістом, а ритм ще узгоджується і рими присутні. Попередній аналіз згенерованого тексту був доповнений краудсорсинговою оцінкою, де користувачі відповідали на опитування щодо відповідних особливостей лірики, створеної попередньою і поточною версією Tra-la-Lyrics. Результати опитування підтвердили покращення текстів пісень, згенерованих системою Tra-la-Lyrics 2.0.

У роботі [15] група авторів описує особливості багатомовного розширення програмної системи PoeTryMe [39] та наводять результати оцінювання генератора віршів. Вони вважають, що створення віршів – це особливий вид генерування природної мови NLG, де зазвичай використовують декілька джерел знань для оброблення функцій на різних рівнях, таких як синтаксис, семантика, форма чи естетика. Хоча це завдання вже було вирішено багатьма попередніми дослідниками [5], [12], [14], [16], [27], [40], [44], [51] і їхні зусилля були націлені на різні мови, однак усі відомі на сьогодні програмні системи зосереджені на обмеженій меті генерування текстів, зазвичай однією мовою. Їхня ж робота зосереджена на адаптації однієї архітектури системи для створення віршів трьома різними мовами – португальською, іспанською та англійською. Наявна архітектура системи спочатку була розроблена та доповнена різними адаптаціями, необхідними для кожної мови, включаючи лінгвістичні ресурси, що використовують для оброблення морфології, синтаксису, семантики та метричного сканування результатів роботи. Автоматичне оцінювання було зроблено так, щоб його можна застосувати до цільових мов. Архітектура системи охоплювала три актуальні принципи створених віршів, а саме: наявність поетичних особливостей, варіативність мовної структури та смисловий зв'язок із заданою темою. Автоматичні заходи, застосовані для другого та третього принципів, можна розглядати як новий підхід до оцінювання згенерованої поезії. Загалом вірші були успішно створені трьома мовами. Незважаючи на незначні відмінності в різних мовах або початкових словах, виявлено, що вірші мають регулярний ритм, часті рими, виявляють цікавий ступінь варіації та семантично асоціюються з початково заданими параметрами.

У роботі [17] автор описує особливості збільшення змістовності в поетичному генеруванні віршів за допомогою Twitter-бота. Водночас, цей самий автор зі своїми колегами в роботі [16] описує особливості генерування інтерактивної поезії за допомогою програмної системи Co-PoeTryMe. Автори вважають, що ця система може автономно генерувати вірші за заданим набором початкових параметрів. Після деякого використання цієї системи багато письменників і інших її користувачів виявили бажання брати участь у творчому процесі її тестування, безпосередньо взаємодіючи з системою PoeTryMe [39]. Їхня робота була мотивована попередніми враженнями від роботи системи. Хоча вона й має деякі проблеми автоматичного генерування віршів, тобто система PoeTryMe має певні обмеження, однак вона повідомляє про нещодавні заходи щодо вирішення анало-

гічних проблем користувачів і надає альтернативні способи використання цієї системи. Передусім, надано деякі функціональні можливості через веб-API, що дало змогу використовувати їх іншими системами. Вона містить засоби для створення окремих рядків, а також пошуку споріднених слів, що дає можливість користувачу встановлювати додаткові обмеження щодо кількості складів, семантику та рими. Окрім цього, веб-API була розроблена спільно з дружнім інтерфейсом системи Co-PoeTryMe, що дає можливість користувачам починати з нуля або зі створеної чернетки. Також в систему була внесена широка можливість редагування тексту, насамперед перекидати рядки та слова або замінювати їх щойно згенерованими альтернативами.

У роботі [51] автори описують особливості розроблення нейронної мережі, яка вивчає англійські тексти, знаходить приховані знання та відповідає на запитання користувачів. Запропоновано нову нейронну мережу, яка може автоматично навчатися та видобувати зміст із текстів й давати відповіді на запитання про зміст будь-якого великого документу або короткий фрагмент тексту.

У роботі [52] автор описує різні способи автоматичної ідентифікації слів, що належать до семантичного поля. Семантичне поле, будучи сукупністю одиниць змісту, охоплює певну область людського досвіду і утворює відносно автономну мікросистему з одним або декількома центрами. Відносини в таких мікросистемах ще називають асоціаціями. Також наведено результати поточних досліджень зі створення семантичного поля поняття "імперія".

У роботі [23] автори описують особливості роботи інструменту Citation AS, призначеного для автоматичного генерування запитань на підставі вмісту цитування, поданого набором деяких речень в оригінальних роботах. Для цього було застосовано лінійний дискримінантний аналіз (LDA) для аналізу розподілу тем змісту цитування. Такий аналіз, будучи окремим типом лінійної комбінації, що описує математичний процес, використовує різні елементи даних і застосовує до цього функції для окремого аналізу декількох класів об'єктів чи елементів. Автори вважають, що їхній інструмент CitationAS може автоматично генерувати всеосяжне, детальне та точне опитування відповідно до пошукових термінів користувача. Це також може допомогти науковцям дізнатися про статус їхнього дослідження в певній галузі знань. Тому цей інструмент є практичним, позаяк об'єднує мітки кластерів із семантичного рівня, щоб покращити результати кластеризації. Інструмент також враховує інформацію про розташування пропозиції під час оцінювання речення за TF-IDF. Нагадаємо, що TF-IDF (від англ. TF – *Term Frequency*, IDF – *Inverse Document Frequency*) – статистичний показник, який використовують для оцінювання важливості слів у контексті документа, що є частиною колекції документів чи корпусу. Вага (значущість) слова пропорційна кількості вживань цього слова у документі й обернено пропорційна частоті вживання слова у інших документах колекції.

У роботі [2] автори описують особливості побудови багатопланової нейронної мережі для короткотермінового прогнозування електричного навантаження у випадку, коли частина вхідних змінних задається в нечисловому вигляді. Запропоновано нову архітектуру нейронної мережі та алгоритми її навчання, які да-

ють можливість виконувати високошвидкісне оброблення інформації, поданої в різних масштабах вимірювань (кількісному, порядковому та номінальному). Типи та параметри використовуваних функцій активації можуть визначатися обсягом наявних попередніх знань. Експериментальне порівняння результатів роботи з традиційною нейронною мережею підтверджує перевагу запропонованого підходу.

У роботі [5] автори описують особливості організації нейронної мережі для оптимізації та оброблення сигналів. Вони вважають, що актуальним завданням такого дослідження є здатність штучних нейронних мереж не тільки вирішувати в режимі онлайн широкий спектр оптимізаційних проблем, а й створювати нові технології та архітектури системи для генерування природної мови NLG. Їхнє дослідження забезпечує поглиблене висвітлення результатів математичного моделювання разом із ілюстративними результатами комп'ютерного моделювання.

У роботі [7] автори описують особливості побудови багатопшарового перцептрона, його архітектури та зворотне поширення помилок. Багатопшаровий перцептрон MLP (англ. *Multilayered Perceptron*), як спрямована штучна нейронна мережа, генерує набір виходів з набору входів, характеризується декількома шарами вхідних вузлів, з'єднаних у вигляді спрямованого графа між вхідним і вихідним шарами. MLP використовує зворотні пропозиції для навчання мережі, також є методом глибокого її навчання.

У роботі [12] автори описують особливості нелінійного моделювання та прогнозування з прямими та рекурентними мережами. Вони вважають, що у мережах із прямим зв'язком сигнали надходять тільки в одному напрямку без зворотного зв'язку. Тому застосування таких мереж у прогнозуванні, обробленні сигналів і управлінні вимагають чіткого оброблення їхньої динаміки. Мережі прямого зв'язку можуть врахувати динаміку, включаючи минулі вхідні та цільові значення в розширений набір вхідних даних. Набагато зручніше динамічне подання можна отримати шляхом врахування у внутрішній мережі зворотного зв'язку. Ці типи моделей називають рекурентними мережевими моделями і їх використовував Джордано (1986) для керування та вивчення плавних рухів роботів, а Елман (1990) – для вивчення та подання тимчасової структури в лінгвістиці. У мережі Джордано минулі значення мережевого виходу повертаються до прихованих одиниць; у мережі Елмана минулі значення прихованих одиниць повертаються в себе.

У роботі [14] автори описують особливості побудови згорткової мережі на підставі регіонів для точного виявлення та сегментації об'єктів. Конволюційна нейронна мережа CNN (англ. *Convolutional Neural Network*), будучи специфічним типом штучної нейронної мережі, використовує перцептрони та алгоритм машинного навчання для аналізу даних. CNN застосовують до оброблення зображень, природних мов й інших видів пізнавальних завдань. Їхній підхід поєднує дві ідеї: можна застосувати згорткові мережі високої пропускної спроможності (CNN) до пропозицій регіону знизу вгору, щоб локалізувати та сегментувати об'єкти; яка позначені навчальні дані мізерні, то контрольована попередня підготовка для допоміжного завдання з подальшим точ-

ним налаштуванням для конкретної області значно підвищує її продуктивність. Оскільки автори поєднали пропозиції регіонів із CNN, то вони назвали отриману модель R-CNN (англ. *Region-Based Convolutional Neural Network*) або згортковою мережею на підставі регіонів [40].

У роботі [27] автори описують особливості побудови рекурентних нейронних мереж для передбачення, алгоритми їхнього навчання, архітектуру та стабільність роботи. Вони вважають, що нові технології в інженерії, фізиці та біомедицині вимагають дедалі складніших методів цифрового оброблення сигналів. Представляючи останню дослідницьку роботу, автори демонструють, як можна реалізувати рекурентні нейронні мережі в реальному часі (RNN), щоб розширити традиційні методи оброблення сигналів і допомогти науковцям боротися з проблемою точного передбачення. У цьому контексті нейронні мережі розглядаються як масово пов'язані між собою нелінійні адаптивні фільтри.

У роботі [40] автори описують особливості побудови швидкої згорткової мережі на підставі регіонів R-CNN (англ. *Region-Based Convolutional Neural Network*), їхній результат на шляху до виявлення об'єктів у реальному часі за допомогою мереж регіональних пропозицій. Вони вважають, що найсучасніші мережі для виявлення об'єктів залежать від алгоритмів пропозицій регіону для підтвердження гіпотези про розташування об'єктів. Такі досягнення, як SPPnet і Fast R-CNN [40], значно скоротили тривалість роботи цих мереж після виявлення об'єктів, вказуючи на обчислення пропозицій регіону як на вузьке місце. Нагадаємо, що мережа SPPnet (англ. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*) – об'єднання просторової піраміди в глибоких згорткових мережах для візуального розпізнавання. Її робота полягає в тому, що CNN є тільки входом фіксованого розміру, оскільки він буде впливати на ефект розпізнавання, а стійкість не підходить для багатомасштабних зображень. SPP-мережа дуже добре вирішує зазначену проблему, позаяк може витягувати об'єкти фіксованого розміру для зображень будь-якого масштабу.

У роботі [49] автори описують особливості організації глобальної стабільності безперервних рекурентних нейронних мереж. Наведено нові результати щодо глобальної асимптотичної стабільності (GAS) та глобальної експоненціальної стабільності (GES) загального класу безперервних рекурентних нейронних мереж з ліпшицевими неперервними та монотонними неспадними функціями активації. Наведено три достатні умови для ГАС нейронних мереж, які можна перевірити, чи відрізняються вони від наявних і чи можна їх покращити. Потім було розширено наявний результат GAS до результату GES, а також розширено наявні результати GES на більш загальні випадки з менш обмежувачими матрицями ваги з'єднання та/або частково функціями активації Ліпшица.

У роботі [50] автори описують особливості побудови алгоритмів навчання для постійного запуску повністю рекурентних нейронних мереж. Ними виведена точна форма алгоритму навчання, що слідує за градієнтом, для повністю рекурентних мереж, що працюють у часі з безперервною вибіркою, і використовують як основу практичні алгоритми для тимчасових контрольованих навчальних завдань. Ці алгоритми мають перевагу, по-

заяк не вимагають точно визначеного інтервалу навчання, які працюють під час роботи мережі. Їхній недолік у тому, що вони вимагають нелокального зв'язку в нейронній мережі, яка навчається, і є дорогими з точки зору обчислень. Ці алгоритми дають можливість нейронним мережам, які мають періодичні з'єднання, вивчати складні завдання, які вимагають збереження інформації протягом певного періоду часу з фіксованою або невизначеною довжиною.

Результати дослідження та їх обговорення

Недавно в мережі Інтернет з'явилося повідомлення [20], що алгоритм штучного інтелекту від компанії Google навчився писати вірші, наслідуючи відомих поетів. Також компанія розробила додаток Verse by Verse, який допомагає творчим особистостям згенерувати вірш у стилі відомих поетів. Текст поезії штучний інтелект генерує на підставі "великої збірки класичної поезії" найвідоміших американських авторів. Користувач може обрати до трьох поетів, зокрема Емілі Дікінсон, Волта Вітмена й Едгара Аллана По. Після того, як користувач обрав структуру свого майбутнього вірша, додаток Verse by Verse попросить скласти перший рядок вірша. Далі алгоритм штучного інтелекту пропонує ще декілька його варіантів. Використовувати надані системою пропозиції вірша необов'язково – їх можна також редагувати або видалити. "Цей інструмент повинен надихати вас, а не створювати від вашого імені цілий вірш – хоча ви можете зробити й це. Після того, як ви доведете рядки до досконалості, ви зможете додати їх у свій майбутній шедевр", – описує додаток Engadget.

1. Особливості віршування та написання пісень.

Художнє мовлення, що є практичним втіленням у віршовані твори, має два основні різновиди: поезію (вірш) і прозу [3], [41], [48]. Під поезією (з грецької – творчість) розуміють ритмічно організоване мовлення, постане на підставі конкретно-історичної версифікаційної системи [10], [25], [26], [34]. Термін вірш (з латинської – повтор, поворот) має декілька значень, одне з яких таке: вірш – це ритмічно організована мова для посилення її виразності й емоційності (звідси його часте використання у ліриці). Взаємозамінність цих термінів проявляється ще й у тому, що віршовані твори певного поета, нації чи епохи називають поезією [9], [11], [48].

Віршована мова характеризується своєрідністю інтонації й особливим темпом, пов'язаним з великою кількістю пауз. У зв'язку з цим питома вага і змістове значення кожного слова в контексті вірша підсилюється [10], [41], [42]. Окрім цього, віршована мова характеризується настановою на музичне звучання [34], [45], [42]. До основних засобів організації вірша належать: ритм, стопа, розмір, рима, строфа (розрізняють ще астрофічний та нерівнострофічний вірші) тощо [11], [25], [26], [43].

Під віршованим твором розуміють зв'язану й послідовну сукупність слів, які розподілені рядами так, що вони слідує певним законам римування [11], [25], [41]. Такі твори найчастіше використовують в літературі та поезії. В літературі віршовані твори мають вигляд віршів, поем, сонет та інших віршованих форм [10], [41], [42].

Вірш – система поетичного мовлення, що має певні закономірності внутрішньої організації ритмічної структури, де особливе значення мають ритмічні акценти [10], [45]. Тому основою, навколо якої будуються дані

поняття, є рима. Віршована рима – це співзвуччя закінчень у словах, які знаходяться на місці клаузул. *Клаузула* – заключна частина віршованого рядка, починаючи з останнього наголошеного складу. У музиці віршовані твори мають вигляд текстів пісень [25], [34], [43]. Основні показники пісні з точки зору віршування – це строфічна будова, повторюваність віршів строфи, розмежування заспіву та приспіву, виразна ритмізація, музичність звучання, проста синтаксична будова [8], [22], [47].

Стопа – найкоротший відрізок певного віршованого метра, сконцентрованого у групі складів з відносно незмінним наголосом (ритмічним акцентом). Стопу сприймають як одиницю виміру та визначення віршованого ритму. В українській силабо-тоніці вона спирається на природне мовне чергування наголошених і ненаголошених складів, які зумовлюють специфіку віршованого розміру [11]. Залежно від кількості складів стопа буває двоскладова (ямб, хорей), трискладова (дактиль, анапест, амфібрахій), подеколи – чотирискладова пеан (хорова пісня-молитва давньогрецької лірики) [22], [41], [45].

Щоб віршований твір мав ритмічне звучання та чітку структуру, рими замало. Навіть сама рима не є настільки важливою, оскільки існують так звані білі вірші, які не мають рими, проте звучать гармонійно і "по-віршованому". Окрім рими, вірш повинен мати ритм. Термін "ритм" означає фактичне розташування наголошених і ненаголошених складів у конкретному віршованому розмірі силабо-тонічної системи, динаміку акцентної ритмо-інтонації у тонічній системі та верлібрі – неримований, нерівно наголошений віршо-рядок.

Римування – особливість розташування рим у вірші, інтервал між ними. У віршованому творі є такі основні види римування [3], [10], [25]:

- паралельне (суміжне, парне) – римують перші два та останні два рядки (ААББ);
- перехресне – римують перший і третій та другий і четвертий рядки (АБАБ);
- кільцеве – римують перший з останнім та другий з третім рядки (АББА);
- потрійне – рідко трапляється, коли римують перші три рядки та останні три (ААА БББ).

Строфою (від грецького – поворот, зміна, коло) називають фонічно завершену віршову сполуку, яка повторюється у поетичному творі, об'єднану здебільшого спільним римуванням, представлена інтонаційною та ритміко-синтаксичною цілісністю, відмежована від аналогічних сполук помітною паузою та іншими чинниками (закінчення римованого ряду, відносна змістова завершеність тощо). Строфа як основна композиційна одиниця, містить декілька рядків, об'єднаних між собою в цикл. Найчастіше строфа як структурна динамічна єдність містить чотири рядки [41]. Римо-ритмічний рисунок у кожній строфі одного вірша майже завжди однаковий, що надає стійкості ритму усього віршованого твору [1], [11], [34]. В цьому якраз і полягає ритмотворче значення строфи.

2. Методи генерування віршованих творів. Отже, автоматичне чи, зазвичай, автоматизоване генерування віршованих творів – не зовсім поширена практика серед поетів чи піснярів. Однак, на програмному ринку останнім часом появилися деякі програмні додатки, які пробають реалізувати такі інтелектуальні завдання [35], [38], [39]. Методи і алгоритми, на яких ґрунтується ро-

бота таких програм, є різними, тому вони й створюють віршовані тексти, зазвичай, не найкращої якості. Загалом, методи генерування віршованих творів поділяють на чотири групи [13]: на підставі шаблонів; генерування та тестування; еволюційні алгоритми; на підставі конкретних випадків.

Загальна ідея *методу на підставі шаблонів* полягає у визначенні набору текстів, слів і дійсних перетворень, які дають змогу автоматично генерувати фіксовану кількість альтернативних варіантів деякого вірша [2], [16]. Віршована структура виходить шляхом виділення слів від його розміру, ритму та виду римування. Потім зі словника вибирають слова, які підходять для синтаксичних і/або ритмічних обмежень, що заповнюють дану структуру. Хоча такий метод і базується на переробленні наявних творів, додаючи до них різні варіації, проте відоме на сьогодні ПЗ генерує доволі змістовні віршовані твори та створює враження досить розумного робота-поета [18], [51], [52].

Робота *методу генерування та тестування* спирається на попередні віршовані тексти та на виборі лексики, наданій користувачем для створення метрично керованої рекомбінації даного словника відповідно до шаблонів ліній, витягнутих з попередніх віршів [2], [14], [18]. Випадкові послідовності слів утворюють відповідно до формальних вимог, які містять різні метрики або обмеження. Наприклад, автоматичний поет WASP [13] використовує набір будівельної евристики, отриманої з формальних метричних обмежень, для створення поеми з множини слів і набору лінійних шаблонів, наданих користувачем. Програмна система використовує метод генерування та тестування шляхом випадкового створення послідовностей слів, які відповідають формальним вимогам побудови віршованого твору [3], [9], [11], [41], [48].

Еволюційні алгоритми є основою комп'ютерних систем, робота яких ґрунтується на виконанні еволюційних обчислень, реалізованих у вигляді нейронної мережі, навченої на даних, отриманих від людських тестувальників [20], [27], [31], [51]. Важливим принципом такого підходу є прийняття реального процесу написання поезії як зразка, з якого має створюватися інтелект комп'ютерної системи. Як відправну точку такий підхід використовує дуже проникливий та інтуїтивний алгоритмічний опис творчого процесу людини під час створення віршованого твору. На підставі цього опису наведена загальна архітектура відповідного програмного додатку, яка має декілька генераторних модулів [16], [23], [51]. Вони створюють певну кількість віршованих творів-кандидатів і модифікують їх структуру на наступних ітераціях. Інші декілька модулів оцінюють їх і вибирають твір-претендент з найвищим рейтингом, заносять його у відповідну базу даних [27], [29], [50]. У ній зберігаються як попередні результати роботи цих модулів, так і відповідні лексикон, різні словоформи, а також концептуальна та синтаксична база знань. Особливість такої архітектури програмного додатку полягає в тому, що модулі оцінювання рейтингу віршованого твору організовані у два рівні – нижній, де відбувається лексичне, граматичне та ритмічне оцінювання кожного варіанту віршованого твору, і верхній, що відповідає за оцінювання логіки та змісту такого твору [28].

Метод *на підставі конкретних випадків* дає можливість створювати віршовані твори на наявних віршах чи

піснях, які потрібно попередньо підібрати відповідно до вхідних даних, введених користувачем. Алгоритм програми передбачає виконання таких кроків [20], [31]:

- 1) з декількох готових прикладів рядків (віршів) витягують конкретний рядок для кожного куплета, що відповідає вхідним даним користувача. Структуру відповідного рядка визначає розподіл передбачуваних вхідних даних – обраної строфічної форми;
- 2) генерують кожен рядок чернетки віршованого твору, що відповідає структурі кожної лінії обраного куплета вірша – іноді у поєднанні зі словами з додаткового джерела лексики, застосовуючи додаткові обмеження для слів;
- 3) відображає чернетку віршованого твору, яку користувачу потрібно підтвердити або виправити;
- 4) проводить аналіз будь-яких готових віршів для додавання відповідної інформації до бази даних, які згодом будуть використані в подальшій роботі.

Наведені методи та алгоритми генерування віршованих творів плануються використати під час розроблення адекватних алгоритмів, який на підставі проаналізованих текстів віршів чи пісень і аналізу їхнього змісту буде генерувати логічні віршовані твори.

3. Обґрунтування методу вирішення завдання. Як було зазначено в п. 1, для реалізації логіки програмного додатку обрано еволюційний алгоритм у поєднанні з методом на підставі конкретних випадків [30]. Сукупно вони дадуть змогу досягти найкращих результатів під час генерування віршованих творів, оскільки еволюційний алгоритм реалізує штучний інтелект, який працює подібно до роботи мозку людини.

Проведені багатьма науковцями дослідження [30], [36], [50] свідчать про те, що людський мозок аналізує попередньо прочитані або почуті вірші чи тексти пісень, запам'ятовує слова, що в них найчастіше вживаються, та загальні настрої, які наповнюють ці твори. Базуючись на цих даних, під час творчого процесу мозок людини "виштовхує" здобуті перед тим знання та знаходить їм унікальне застосування. Саме для того, щоб реалізувати аналітичну частину роботи мозку, було дещо модифіковано метод на підставі конкретних випадків [2], [20], [36], який зберігає оригінали готових віршованих творів і використовує їх для генерування нових. Відмінність запропонованого методу від стандартного методу полягає в тому, що в базі даних будуть зберігатися не цілі віршовані твори і навіть не їх частини, а зв'язки між словами у рядках куплета віршованого твору.

Для реалізації еволюційного алгоритму [20] потрібно розробити нейронну мережу, яка має аналізувати запропоновані користувачем віршовані твори та визначати слова, які найчастіше трапляються разом. Враховуючи частоту появи цих слів і їхню належність до певних змістових груп, нейронна мережа буде створювати послідовність слів, логічно зв'язаних між собою завдяки попередньо проаналізованим даним.

Завдання створення нейронної мережі. Нейронна мережа – це послідовність нейронів, з'єднаних між собою синапсами [31]. Це машинна інтерпретація роботи мозку людини, яка здатна сама навчатися та видавати результати на підставі проаналізованих даних. Отже, завдання створення нейронної мережі означає побудову нейронів і зв'язків між ними (синапсів). На рис. 1 показано базову схему нейронної мережі, де І – вхідний нейрон (Input), Н – прихований нейрон (Hidden), О – вихідний нейрон (Output).

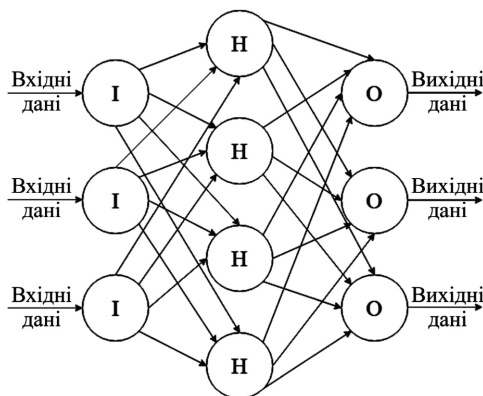


Рис. 1. Схема нейронної мережі

Під нейроном будемо розуміти обчислювальну одиницю, яка отримує дані, проводить над ними прості обчислення та передає їх далі. Нейрони бувають вхідні, приховані та вихідні [31]. Вхідний нейрон отримує дані, деяка множина прихованих нейронів їх обробляє, а вихідний нейрон виводить результат розрахунку. У тому випадку, коли нейронна мережа містить велику кількість нейронів, вводять термін нейронного шару. У кожного з нейронів є два основні параметри: вхідні дані (input data) і вихідні дані (output data). У разі вхідного нейрону, то маємо $input = output$. В інших випадках у поле input потрапляють дані від всіх нейронів з попереднього шару, після чого їх нормалізують за допомогою функції активації $F_{act}(x)$ і отриманий результат потрапляє в поле output.

Синапс – це зв'язок між двома нейронами. У синапсів є параметр ваги, завдяки якому, вхідні дані змінюються, коли їх передають від одного нейрона до іншого.

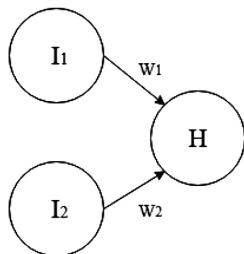


Рис. 2. Синапси з їхньою вагою

На рис. 2 зображено приклад роботи прихованого нейрона Н, де дані визначають за формулою:

$$H_{input} = I_1 \cdot W_1 + I_2 \cdot W_2. \quad (1)$$

Вихідні дані визначають функцією активації в такому вигляді:

$$H_{output} = F_{act}(H_{input}). \quad (2)$$

Щоб отримати остаточне значення, застосовують функцію активації до отриманих сум прихованих шарів. Така функція перетворює вхідний сигнал у вихідний шляхом нормалізації вхідних даних. Залежно від потреб розрахунку в різних прихованих нейронів функції активації бувають різні, а саме:

- лінійна функція: $F_{act}(x) = x$; (3)

- сигмоїдна: $F_{act}(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$; (4)

- гіперболічний тангенс: $F_{act}(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$. (5)

Лінійна функція не є достатньо ефективною, а гіперболічний тангенс застосовують тоді, коли присутні як додатні, так і від'ємні значення даних. Отже, в цій робо-

ті спробуємо використати сигмоїдну функцію. Нагадаємо, сигмоїда – це неперервно диференційована монотонна нелінійна S-подібна функція, яку часто застосовують для "згладжування" значень деякої величини. Часто під сигмоїдою розуміють логістичну криву.

Навчання нейронної мережі. Щоб нейронна мережа видавала правильні результати, її потрібно навчити. Для навчання нейронної мережі програмного додатку було вибрано метод зворотного поширення. (англ. *Backpropagation*). Це ітеративний градієнтний алгоритм, який використовують для мінімізації помилки роботи багатоваріового перцептрону та отримання бажаного результату на виході нейронної мережі. Основна ідея цього методу полягає в поширенні сигналів помилки від виходів мережі до її входів, тобто в напрямку, зворотному прямому поширенню сигналів у звичайному режимі роботи [29].

Спочатку потрібно визначити, у чому саме полягає сигнал помилки, потім потрібно відкоригувати ваги зв'язків між нейронами, щоб зменшити вагомість помилок (рис. 3). Обчислення змін до цих ваг проходить у два етапи:

- 1) Спочатку знаходять вихідну суму похибки (яку отримано після застосування функції активації), щоб відкинути необхідну зміну вихідної суми (зазвичай, називають дельта-виходом)
- 2) Потім витягують зміну ваг шляхом множення дельта-вихідної суми на значення прихованого шару.

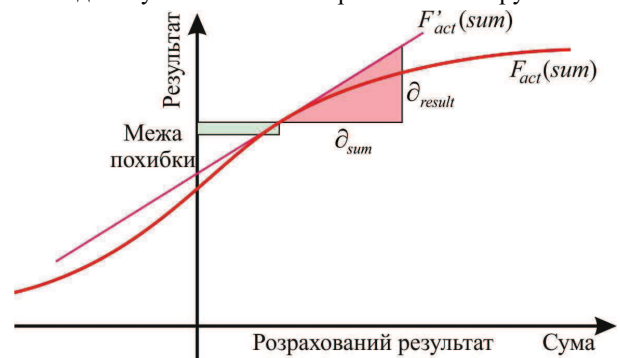


Рис. 3. Графік сигмоїдної функції та її похідної

Вихідну суму сигналу похибки обчислюють як різницю між очікуваним і розрахованим результатом, а саме:

$$Err_{output\ margin\ sum} = target - calculated. \quad (6)$$

Похідна від сигмоїди (4) дасть швидкість зміни (або кут "нахилу") функції активації на вихідній сумі, зокрема

$$F'_{act}(sum) = \frac{\partial H_{sum}}{\partial H_{result}}. \quad (7)$$

Оскільки вихідна сума сигналу похибки (6) є різницею в отриманому результаті, то можна помножити її на швидкість зміни, щоб отримати суму різниці виходу:

$$\frac{\partial H_{sum}}{\partial H_{result}} \cdot Err_{output\ margin\ sum} = \Delta sum. \quad (8)$$

У такий спосіб розраховують запропоновану зміну в сумі вихідного шару. На рис. 3 зображено графік сигмоїдної функції та її похідної, щоб мати наочне уявлення про попередні розрахунки.

Тепер визначимо нову зміну ваг синапсів. Для початку нагадаємо, що вихідний результат є добутком результату прихованого нейрона та ваги синапса між прихованим і вихідним нейронами:

$$H_{result} \cdot W_{H \rightarrow O} = O_{sum}. \quad (9)$$

Похідна від вихідного результату (9) має вигляд:

$$\frac{\partial O_{sum}}{\partial W_{H \rightarrow O}} = H_{result} \cdot \quad (10)$$

Це можна відобразити як

$$\partial W_{H \rightarrow O} = \frac{\partial O_{sum}}{H_{result}} \cdot \quad (11)$$

Завдяки цьому виразу можна знайти нові ваги для синапсів між вхідним і прихованим нейронами. Для цього потрібно спочатку знайти похідну від вхідної суми прихованого нейрона:

$$\partial H_{sum} = \frac{\partial O_{sum}}{W_{H \rightarrow O}} \cdot F'_{act}(sum) \cdot \quad (12)$$

Вхідна сума прихованого нейрона H_{sum} є добутком вхідного нейрона на ваги синапсу між вхідним і прихованим нейронами:

$$I \cdot W_{I \rightarrow H} = H_{sum} \cdot \quad (13)$$

Звідси виводимо, що:

$$\frac{\partial H_{sum}}{\partial W_{I \rightarrow H}} = I, \quad \partial W_{I \rightarrow H} = \frac{\partial H_{sum}}{I} \cdot \quad (14)$$

У такий спосіб можна розрахувати нові значення ваг синапсів між всіма нейронами. Маючи покращені значення ваг, вхідні сигнали помилок будуть допущені з меншою похибкою. Якщо повторювати цей алгоритм ітеративно, то значення таких помилок будуть асимптотично наближатися до нуля.

Основні властивості еволюційних алгоритмів.

Еволюційний алгоритм є багатоточковим стохастичним алгоритмом ітераційного пошуку [30]. Це так звана форма евристичного пошуку, яка одночасно досліджує декілька точок у просторі пошуку і стохастично переміщується у пошуковому просторі, щоб запобігти досягненню локальних максимумів.

Всі еволюційні алгоритми підтримують популяцію осіб $x_i^{(t)}$ протягом часу t в такий спосіб:

$$P(t) = \{x_i^{(t)}, i = \overline{1, n}, t = \overline{1, T}\} \cdot \quad (15)$$

Кожна індивідуальна особа $x_i^{(t)}$ представляє потенційне вирішення поставленої проблеми, яку можна реалізувати як деяку складну структуру даних. Основні кроки еволюційного алгоритму полягають в такому:

- 1) **Ініціалізація.** Для початкового часу $t = 0$ потрібно побудувати нову популяцію $P(t)$, що представляє набір вихідних точок для вивчення простору пошуку. В ідеалі набір точок рівномірно розподілений по всьому простору.
- 2) **Відбір популяції.** Нову популяцію $P(t+1)$ формують шляхом стохастичного відбору індивідів від попередньої популяції $P(t)$, як правило, з ухилом до більш відповідних індивідів. Під час такого вибору кількості осіб $x_i^{(t)}$ їх обирають з поточної популяції, нащадки яких будуть присутні в наступному поколінні.
- 3) **Оцінювання.** Кожну поточну особу $x_i^{(t)}$ оцінюють, щоб встановити деяку міру її "придатності". Під час такого оцінювання застосовують функцію оптимізації до кожного індивіда рішення $x_i^{(t)}$, що дає змогу отримати числову оцінку, яка вказує на її придатність для вирішення проблеми. Функцію оптимізації можна реалізувати як агрегацію набору окремих функцій, які є оцінювачами, кожен з яких призначений для вимірювання іншої властивості або особливості цього індивіда. Така багаточільова природа оптимізації вводить проблему інтеграції всіх балів, що повертаються з кожного оцінювача. Одним з потенційних рішень є їхня агрегація в лінійній комбінації.

4) **Еволюція рішень.** Деякі члени нової популяції проходять трансформацію за допомогою "генетичних операторів" для формування нових рішень. Еволюційний алгоритм має набір таких операторів, які є функціями для зміни рішення, що дає змогу іншим рішенням викликати переміщення в просторі пошуку. Ці оператори є або одинарними функціями m_i (мутаторами), які викликають випадково більш високі функції перехрещування c_i (crossover), що намагаються поєднати властивості двох або більше рішень.

5) **Повторюються кроки** від 2 до 4 до завершення ітераційного процесу. Його припинення настає або після того, як пройшла певна кількість ітерацій, або знайдено чи не знайдено оптимальне рішення.

Після припинення роботи еволюційного алгоритму очікується, що знайдене рішення є оптимальним або близьким до нього (рис. 4).

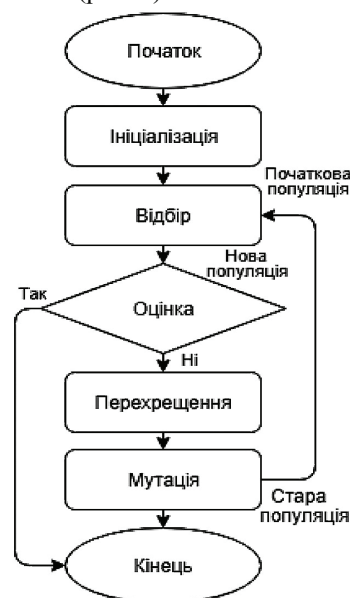


Рис. 4. Схема роботи еволюційного алгоритму

Використання генетичного алгоритму для генерування віршованих творів. Генетичний алгоритм – одна з реалізацій еволюційних алгоритмів для розв'язання задач оптимізації шляхом послідовного підбору, комбінування та варіанти шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію. Особливістю генетичного алгоритму є зосередження уваги на використанні операторів схрещення, які виконують операцію рекомбінації рішень-кандидатів, роль якої аналогічна ролі схрещення в живій природі і мутації, яка виводить нові властивості рішень-кандидатів після їхньої перевірки та оцінювання. Якщо розглядати генетичний алгоритм з точки зору навчання нейронної мережі, то можна виділити такі етапи навчання:

- отримують вихідну популяцію з множини рішень, випадково створених нейронною мережею;
- генерують множину віршованих творів на підставі рішень з вхідної популяції;
- обчислюють функцію пристосованості кожного рішення для оцінювання його якості;
- оцінюють поточне покоління для створення нового за допомогою генетичних операторів через стратегію заміни.
- повторення алгоритму проходить доти, поки не буде досягнуто очікуваний результат.

Оскільки популяція повинна еволюціонувати з найкращих рішень, то необхідно визначити функцію прис-

тосованості (англ. *Fitness Function*) – це метрика, що вимірює якість об'єкта. У цій роботі функцією пристосованості слугуватиме оцінка користувача, яку він ставитиме на його розсуд. Відповідно до виставленої оцінки буде відібрано відповідні рішення, яке виявиться найкращим. Отже, для нашого випадку функція пристосованості – це показник змістовності віршованого тексту на думку користувача.

Як доповнення до генетичного алгоритму, наведемо етапи застосування природної еволюції до вмираючого покоління. Зазвичай, виживають кращі рішення, а їхні нащадки замінюють найгірші об'єкти в такий спосіб:

- упорядковують рішення поточного покоління за рівнем їх пристосованості;
- вибирають n_{wi} кращих рішень (переможців) і передають їх безпосередньо в наступне покоління;
- створюють одного нащадка як результат перехрещення між двома найкращими переможцями;
- створюють $n_{wi} - 1$ нащадків як результати перехрещення двох випадкових переможців;
- створюють $n - n_{wi}$ нащадків як прямі копії двох випадкових переможців;
- застосовують до кожного нащадка випадкові мутації, щоб додати варіативність.

Мутація у випадку генерування змістовних віршованих текстів – випадкова зміна однієї або декількох способів вибору наступного слова. Відомі мутація конкретного значення та мутація за щільністю [36].

Мутація конкретного значення (англ. *Real Valued Mutation*), де спочатку вибирають величину кроку мутації – числова характеристика зміни властивостей рішення. Найбільш ефективні маленькі кроки, але великі кроки значно пришвидшують процедуру розрахунку. Властивості рішень (гени) можуть мутувати за таким правилом:

$$g_{new} = g_{old} \pm \alpha \cdot \delta, \quad (16)$$

де: g_{new} – новий ген; g_{old} – старий ген; знак + або – вибирають випадково; $\alpha = 0,5$ – пошуковий простір; показник δ визначають за формулою

$$\delta = \sum_{i=1}^m \alpha(i) \cdot 2^{-i}, \quad (17)$$

де: $\alpha(i) = 1$ з ймовірністю p ; m – величина кроку мутації.

Мутація за щільністю (англ. *Density Mutation*) – стратегія мутації кожного гена нащадка із заданою ймовірністю. Тобто, окрім ймовірності застосування мутації до самого нащадка також використовують ще ймовірність застосування мутації до кожного його гена, величину якої вибирають з таким розрахунком, щоб в середньому мутували від 1 до 10% генів. Відбувається це в такий спосіб:

- 1) Приєднання випадкового гена в кінець послідовності – з сукупності всіх можливих значень генів до кінця послідовності додають випадковий ген g_{new} : $G \rightarrow \{g_1, \dots, g_m, g_{new}\}$.
- 2) Приєднання випадкового гена в будь-яку позицію послідовності – вставка випадкового гена g_{new} з сукупності всіх можливих значень генів у випадково обрану позицію в послідовності: $G \rightarrow \{g_1, \dots, g_i, g_{new}, g_{i+1}, \dots, g_n\}$.
- 3) Видалення випадкового гена з послідовності – видалення випадкового обраного гена g_i з послідовності: $G \rightarrow \{g_1, \dots, g_{i-1}, g_{i+1}, \dots, g_n\}$.
- 4) Обмін генами в послідовності – обмін місцями в послідовності двох сусідів одного випадково обраного гена g_i : $G \rightarrow \{g_1, \dots, g_{i+1}, g_i, g_{i-1}, \dots, g_n\}$.

4. Алгоритм генерування віршованих творів програмним додатком "GeneratorPC". Проаналізувавши методи генерування віршованих творів, наведених у п. 2, було вирішено здійснити поєднання еволюційного алгоритму та методу на підставі конкретних випадків для досягнення поставлених цілей.

Метод на підставі конкретних випадків полягає в аналізі поданих користувачем віршованих творів. Згенеровані нові віршовані тексти будуть базуватися на попередньо проаналізованих зразках, але не будуть повторювати їхню структуру та загальний зміст. Тут послідовність кроків є такою:

- 1) Користувач задає віршований твір для аналізу.
- 2) Визначають структуру віршованого твору.
- 3) Структура віршованого твору та нові слова записують в базу даних.
- 4) Аналізують слова віршованого твору.
- 5) За допомогою нейронної мережі аналізують зв'язки між словами.
- 6) Проаналізовані зв'язки демонструють користувачеві.
- 7) Якщо користувачеві не подобається наведений результат, він може внести правки.
- 8) Правки враховують і методом зворотного поширення тренують нейронну мережу.

Як еволюційний алгоритм у роботі використано його різновид, а саме – генетичний алгоритм, який застосують для навчання нейронної мережі шляхом оптимізації множин рішень з використанням операцій перехрещення та мутації. Послідовність дій процедури генерування віршованих творів генетичним алгоритмом є такою:

- 1) Користувач задає параметри віршованого твору.
- 2) Вхідні параметри аналізують та відправляють на оброблення до нейронної мережі.
- 3) Нейронна мережа генерує логічно зв'язані тексти за допомогою збережених у базі даних зв'язків між віршованими текстами та вагами синапсів.
- 4) Слова з отриманих віршованих текстів розташовують так, щоб вони відповідали правилам написання віршів.
- 5) Результат подають користувачеві для аналізу та прийняття поточного чи остаточного рішення.
- 6) Якщо користувачеві не подобається результат, він може внести правки.
- 7) Правки враховують і нейронну мережу тренують генетичним алгоритмом.

Як видно з рис. 5 та рис. 6, обидва алгоритми передбачають розроблення нейронної мережі, яка аналізуватиме запропоновані користувачем віршовані твори та генеруватиме зв'язки як між словами, так і рядками куплета вірша. Нейронна мережа буде вдосконалюватися шляхом її навчання методами зворотного поширення та генетичним алгоритмом.

Щоб забезпечити дотримання всіх правил віршування буде розроблено відповідний алгоритм, який стежитиме за структурою віршованого твору, контролюватиме дотримання правил побудови строф, ритму та віршованого розміру. Цей алгоритм також розставлятиме згенеровані нейронною мережею слова в такому порядку, щоб вони правильно римувався та звучали.

5. Специфікація вимог до програмного додатку "GeneratorPC". У проекті спробуємо розробити програмний додаток для автоматизованого генерування віршованих творів. Програмний продукт використовуватимуть поети і піснярі для полегшення творчої роботи над текстами віршів чи пісень.

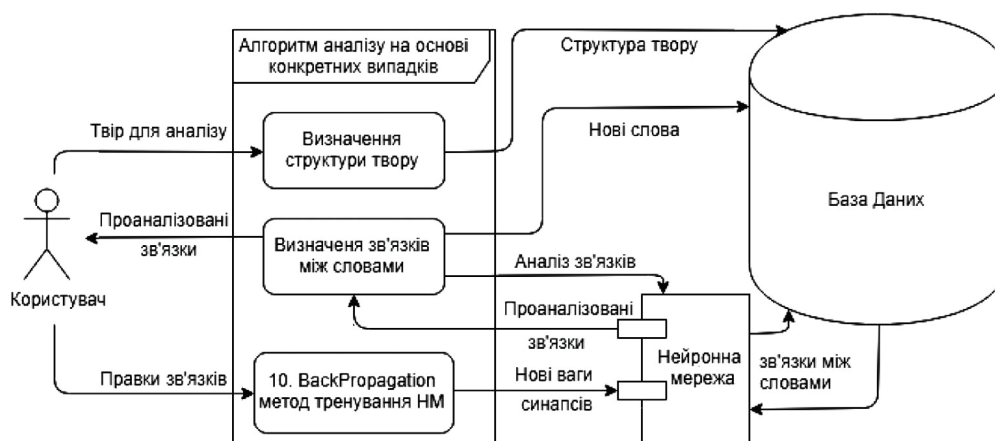


Рис. 5. Алгоритм аналізу віршованих творів

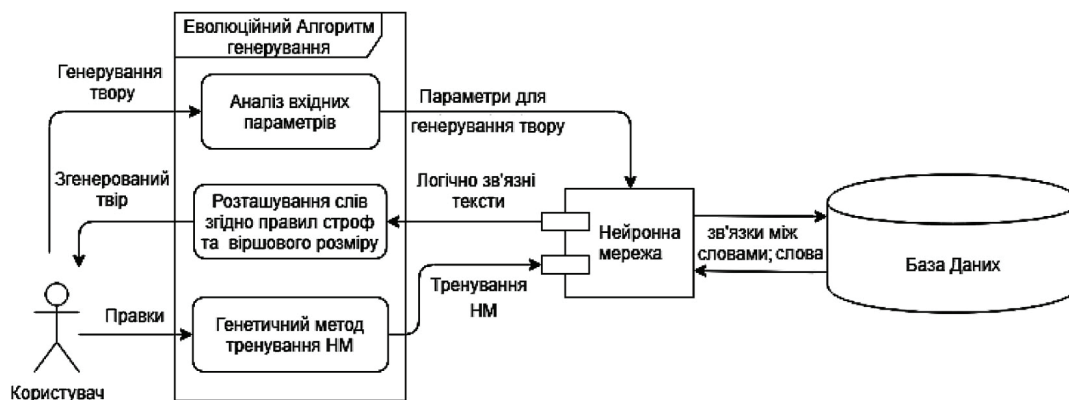


Рис. 6. Алгоритм генерування віршованих творів

Функціональні вимоги до програмного додатку "GeneratorPC" є такими:

- 1) Програмний додаток має надати користувачу інтерфейс у вигляді веб-клієнта для ефективної взаємодії з ним.
- 2) Програмний додаток має генерувати початкові віршовані твори з усіма обмеженнями, які стосуються їх написання (рима, ритм, та ін.)
- 3) Кожен користувач отримує свою власну початкову базу даних, яку він згодом може наповнити проаналізованими текстами, якими сам забажає.
- 4) Віршований твір, згенерований програмним додатком, має відповідати всім вимогам, щоб його назвали віршем: має мати чітку структуру, закономірне римування, віршований розмір, який можна простежити протягом усього віршованого твору.
- 5) Віршований твір, згенерований програмним додатком, має мати логічний зміст і не бути хаотичним набором слів, які не пов'язані між собою.
- 6) Користувач має мати можливість задати структуру віршованого твору: закономірність римування, кількість рядків у куплетах, вид, віршований розмір.

Вимоги до користувацького інтерфейсу:

- 1) Інтерфейс користувача має бути у вигляді веб-сторінки.
- 2) Інтерфейс користувача має бути інтуїтивно зрозумілий та простий у використанні.
- 3) Навігація сторінками не має порушувати "правило трьох кліків", тобто на будь-яку сторінку можна потрапити не більше ніж трьома переходами.
- 4) Інтерфейс користувача має надати такі сторінки взаємодії:

- домашня сторінка;
- сторінка реєстрації та авторизації;
- редактор віршованих творів;

- аналізатор віршованих творів;
- сторінка допомоги та про програму.

Навігацію клієнтською частиною програмного додатку потрібно спроектувати так, щоб забезпечити користувачам можливість перейти до необхідної їм сторінки без зайвих дій. Потрапивши на головну сторінку, користувач має мати можливість авторизуватися, зареєструватися або переглянути інформацію про програму. Після авторизації користувач побачить свою особисту сторінку, з якої він матиме змогу перейти до сторінок де можна генерувати віршовані твори, їх аналізувати або переглянути інформацію про роботу нейронної мережі (рис. 7).

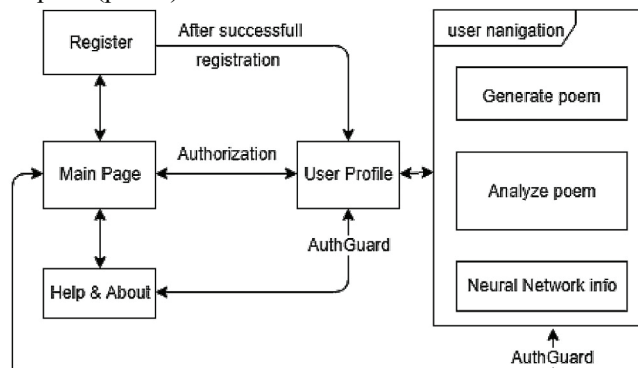


Рис. 7. Навігація по веб-сторінках

З технічної точки зору, користувачів програмного додатку поділимо на авторизованих і неавторизованих. Обидва види користувачів можуть генерувати віршовані твори, при цьому можна вказувати ключові слова, задавати свою структуру віршованого твору, або допов-

нювати наявні твори. Різниця між ними полягає в тому, що авторизований користувач після реєстрації та оплати за підписку отримує власний профіль, до якого створюють окрему базу даних з початковим набором слів. Позначення "AuthGuard" на рис. 7 означає, що користувач матиме доступ до цих сторінок тільки тоді, якщо він авторизований.

Апаратні вимоги до програмного додатку:

- 1) Програма має працювати на комп'ютерах з частотою процесора не менше 1.8 ГГц та оперативною пам'яттю не менше 1 Гб.
- 2) Програмний додаток має працювати коректно на всіх сучасних браузерах.

Класи користувачів програмного додатку "GeneratorPC" розробимо для полегшення творчої роботи над текстами віршів чи пісень для поетів і піснярів, зазвичай початківців чи у випадку творчого їх занепаду (рис. 8).

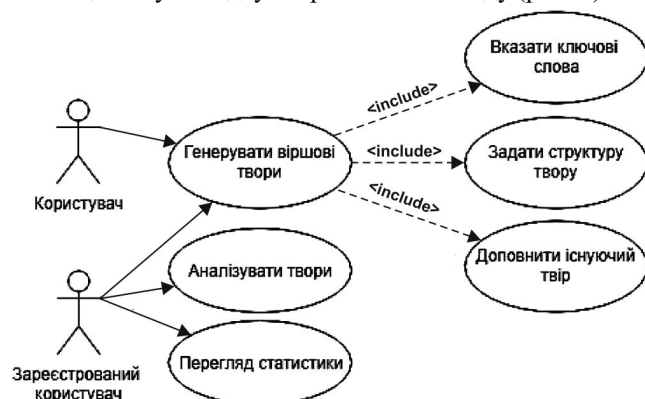


Рис. 8. Діаграма прецедентів

Користувач може здійснювати аналіз наявних віршованих творів і в такий спосіб наповнювати свою базу даних словами і словосполученнями, якими він вважає за потрібне. Також авторизований користувач може переглянути статистику своєї діяльності або інформацію про свою базу даних і нейронну мережу (див. рис. 8).

Обговорення результатів дослідження. Проаналізуємо деякі наявні програмні додатки для генерування віршованих творів, які поєднують наведені в п. 2 методи їхньої роботи, а також запропонований у п. 4 алгоритм генерування віршованих творів програмним додатком "GeneratorPC".

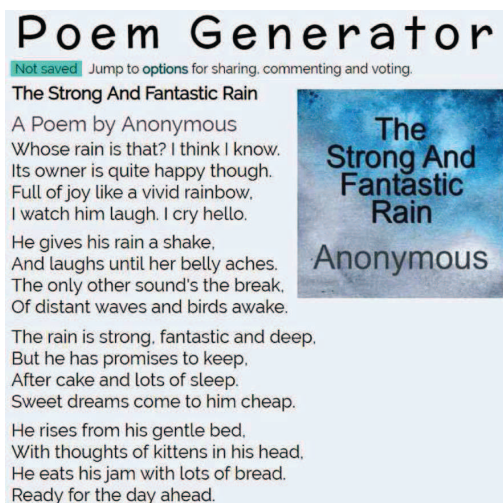


Рис. 9. Результат роботи Poem Generator [38]

Poem Generator – англomовний ресурс, генерує віршовані твори, поєднуючи метод на підставі шаблонів і

метод на підставі конкретних випадків [38]. Його робота полягає в тому, що в систему початково закладено певні шаблони для різних типів віршів. Система застосовує вхідні дані, вказані користувачем наявні шаблони і внаслідок їхнього аналізу генерує відповідний віршований твір.

Переваги ресурсу в тому, що користувач може сам вибирати зміст віршованого твору, задавати певні ключові слова, які будуть фігурувати у ньому. Його недоліки – дуже одноманітні та схожі один на одного результати роботи. Навіть при введенні різних вхідних даних, згенеровані тексти мають подібну структуру та навіть можуть мати однакові рядки (рис. 9).

Стихоробот – російськомовний ресурс [38], принцип роботи якого нагадує метод генерування та тестування, але реалізований не повною мірою. Зазвичай система просто перемішує рядки наявних віршованих творів так, щоб їхні віршовані розміри та рима співпадали. Також цей ресурс дає змогу доповнювати запропоновані тексти наявними в базі даних рядками зі вже згенерованих віршованих творів (рис. 10).

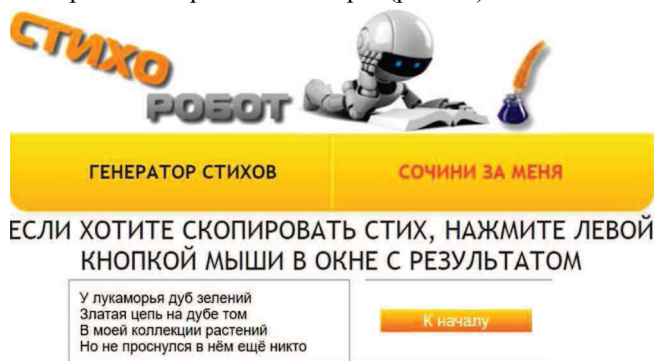


Рис. 10. Результат роботи Стихоробот [38]

Оскільки система використовує готові рядки з наявних віршів, то згенерований віршований текст має зміст хоча б одного рядка, а якщо поталанить, то можна отримати змістовний ланцюжок на декілька рядків чи навіть на куплет вірша. Однак, цей ресурс не містить новизни згенерованого тексту, оскільки він складається із уривків наявних віршованих творів.

PoeTryMe – інтелектуальна платформа для автоматичного генерування віршованих творів із універсальною архітектурою (рис. 11), яка забезпечує значні можливості налаштування відповідних опцій [35], [39]. Перед створенням вірша користувач може визначити багато параметрів, у т.ч. форму вірша, шаблони рядків, доступні семантичні знання, початкові слова, які визначають область генерування, передані настрої та стратегію побудови віршованих творів логічної структури. Реалізація системи поєднує метод генерування та тестування з еволюційним алгоритмом. Його основними складовими є [35]:

- семантична мережа зі словами та семантичними відношеннями;
- граматики з лінійними шаблонами, які є семантичними відношеннями відтворення;
- генератор рядків віршованого твору заданої структури та логічного змісту;
- інструмент для виконання операцій, пов'язаних зі складаннями віршованого рядка;
- контекстуалізатор для пояснення вибору слів і рядків до віршованого твору.

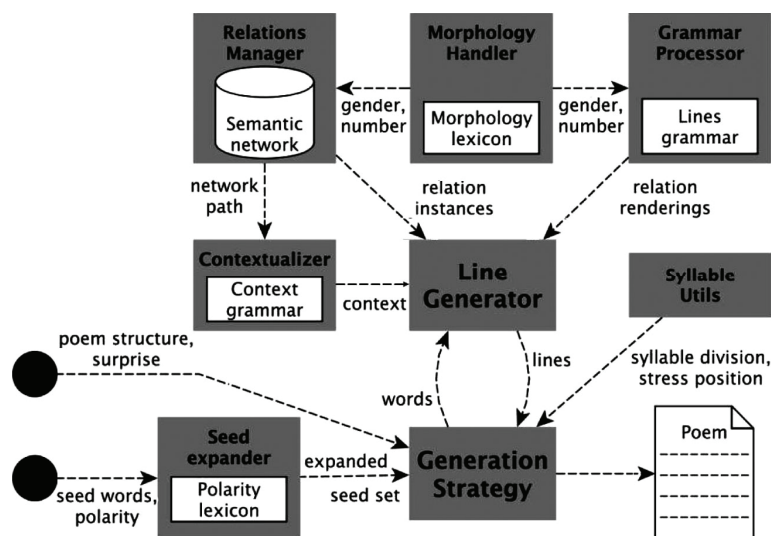


Рис. 11. Архітектура системи PoeTryMe [35]

Згенеровані системою віршовані тексти є неповторними, в ній можна вибирати віршовані розміри, додавати ключові слова, вибирати шаблони віршів (рис. 12). Однак, доволі часто система генерує безглузді словосполучення, в яких немає ніякого змісту, а також дуже тривале генерування. Наприклад, 4 рядки куплета вірша система PoeTryMe генерує приблизно 2 хв.

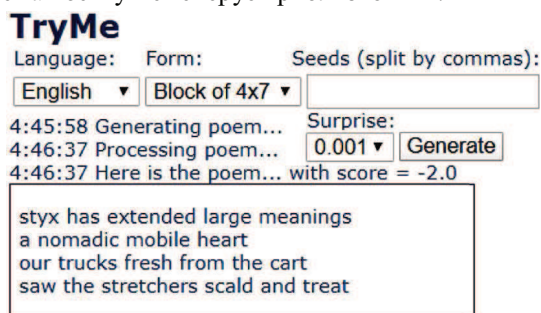


Рис. 12. Результат роботи системи PoeTryMe [39]

Отже, на підставі проведеного аналізу прототипів наявних продуктів потрібно розробити метод генерування віршованих творів, реалізувати його у вигляді компонент програмного додатку "GeneratorPC", порівняти отримані результати його роботи із результатами роботи наявних програмних продуктів, оцінити практичну значущість та шляхи впровадження отриманих результатів. Основні відмінності такого програмного додатку від перерахованих вище полягатимуть в такому:

- поєднання еволюційного алгоритму та методу на підставі конкретних випадків;
- програма може доповнювати запропоновані користувачем віршовані твори;
- можливість створення приватного акаунту, який дасть змогу авторизованому користувачу мати широкі можливості роботи порівняно з неавторизованими;
- користувач може заповнювати свою особисту базу даних словами і словосполученнями, якими він хоче генерувати віршовані твори.

Висновки

1. З'ясовано, що автоматизоване генерування віршованих творів є складною й творчою роботою навіть для професійних поетів і піснярів, яка полягає в написанні текстів віршів і пісень. Обґрунтовано актуальність роботи, наведено об'єкт і предмет дослідження, сформу-

льовано мету роботи та основні завдання дослідження, визначено їхню наукову новизну та практичну значущість отриманих результатів.

2. Проаналізовано відомі літературні джерела та наявні програмні продукти, які можуть генерувати віршовані твори, поєднуючи різні методи та алгоритми. Встановлено, що жоден з них не здатен забезпечити змістовність і унікальність віршованого твору водночас, тим більше українською мовою. Викладено ідеї та робочі гіпотези щодо вирішення прикладної проблеми автоматизованого генерування логічно зв'язаних словесних структур та побудови з них віршованих творів.

3. Проаналізовано наявні підходи до генерування віршованих творів, серед яких актуальними є метод на підставі шаблонів, генерування та тестування, еволюційні алгоритми та метод на підставі конкретних випадків. Досліджено особливості генерування віршованих творів, насамперед правила римування, види строф, віршовані ритми та розміри. З'ясовано, що розроблений програмний додаток має дотримуватися всіх зазначених правил, а також має забезпечити його заданий зміст і логічну структуру.

4. Розроблено підхід до вирішення проблеми, а саме – метод автоматизованого генерування віршованих творів з використанням еволюційних алгоритмів і методу на підставі конкретних випадків. Встановлено, що їхнє поєднання нагадує послідовність дій для творчих особистостей під час створення віршів або написання текстів пісень. Розглянуто призначення та межі застосування майбутнього програмного додатку, який передбачає систему із чотирьох основних елементів – бази даних, нейронної мережі, серверної та клієнтської частин.

5. Розглянуто особливості організації нейронної мережі для автоматизованого генерування віршованих творів. З'ясовано, що навчання нейронної мережі має відбуватися за методом зворотного поширення та з використанням генетичного алгоритму. Проаналізовано принцип роботи алгоритмів пошуку оптимальних рішень, які містять такі послідовні етапи як ініціалізацію, оцінювання рішень, відбір популяцій, еволюцію рішень. Встановлено, що еволюційні алгоритми використовують генетичні операції перехрещення та мутації, призначені для вдосконалення множини можливих рішень. Детально досліджено їхню взаємодію та різні можливості для навчання нейронної мережі.

6. Розроблено алгоритм, за яким програмний додаток "GeneratorPC" буде аналізувати запропоновані користувачем віршовані твори та генерувати нові його варіанти на підставі отриманих від нейронної мережі логічно зв'язаних слів чи рядків куплета вірша. Користувач може вносити правки як до складових вірша, так і до згенерованих віршованих творів, і в такий спосіб може навчати нейронну мережу. Розроблено специфікацію вимог до програмного додатку "GeneratorPC", визначено функціональні вимоги, вимоги до користувацького інтерфейсу та апаратні вимоги, а також визначено потенційні класи користувачів, які будуть його використовувати.

References

- [1] Belekova, L. I. (2009). Syntactic organization of texts of modern American poetry: cognitive-semiotic and linguo-synergetic aspects. *Bulletin of VN Karazin Kharkiv National University*, 57(838), 20–28. Kharkiv: KhNU Publishing House. [In Ukrainian].
- [2] Bodyanskiy, Ya., Popov, S., & Rybalchenko, T. (2008). Multilayer neuro-fuzzy network for short term electric load forecasting. *Lecture Notes in Computer Science – Berlin, Heidelberg: Springer-Verlag*, 5010, 339–348. Retrieved from: https://link.springer.com/chapter/10.1007/978-3-540-79709-8_34
- [3] Brick, O. M. (1927). Rhythm and syntax. *New Lef*, 3–6, 15–37. Moscow: Gosizdat. [In Russian].
- [4] Caza. (2021). Synaptic.js. The javascript architecture-free neural network library for node.js and the browser. Retrieved from: <https://caza.la/synaptic/#/>.
- [5] Cichocki, A., & Unbehauen, R. (1993). *Neural Networks for Optimization and Signal Processing*. Stuttgart: Teubner, 526 p. Retrieved from: <https://www.amazon.com/Neural-Networks-Optimization-Signal-Processing/dp/0471930105>
- [6] Diagrams.net. (2021). Draw.io. Retrieved from: <https://about.draw.io/about-us/>
- [7] Du, K.-L., & Swamy, M. N. S. (2014). Multilayer Perceptrons: Architecture and Error Backpropagation. *Neural Networks and Statistical Learning*, 83–126. https://doi.org/10.1007/978-1-4471-5571-3_4
- [8] Eichenbaum, B.M. (1987). The theory of the "Formal Method". About literature. Moscow: Sov. Writer, 375–408. [In Russian].
- [9] Gasparov, M. L. (1985). Opposition "verse-prose" and the formation of Russian literary verse. *Russian versification: Traditions and development problems*, 4, 264–277. Moscow. [In Russian].
- [10] Gasparov, M. L. (1994). Linguistics of verse. *Izvestia RAN. Literature and Language Series*, 53(6), 28–35. Moscow. [In Russian].
- [11] Gasparov, M. L. (2001). Verb rhyme and the syntax of a poetic line. *Russian language in scientific coverage*, 1, 148–160. Moscow. [In Russian].
- [12] Geqay, R., & Liu, T. (1997). Nonlinear modeling and prediction with feed forward and recurrent networks. *Physica D*, 108, 119–134. [https://doi.org/10.1016/S0167-2789\(97\)82009-X](https://doi.org/10.1016/S0167-2789(97)82009-X)
- [13] Gervas, P. (2002). Exploring Quantitative Evaluations of the Creativity of Automatic Poets. Pablo Gervas. 15th European Conference on Artificial Intelligence. Retrieved from: <http://nil.fdi.ucm.es/sites/default/files/GervasECAIws2002.pdf>.
- [14] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2016). Region-Based Convolution Networks for Accurate Object Detection and Segmentation. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1), 142–158. <https://doi.org/10.1109/TPAMI.2015.2437384>
- [15] Hugo Gonalo Oliveira, Raquel Hervás, Alberto Díaz & Pablo Gervás. (2017). Multilanguage Extension and Evaluation of a Poetry Generator. In *Journal of Natural Language Engineering*, 23(6), 929–967. <https://doi.org/10.1017/S1351324917000171>
- [16] Hugo Gonalo Oliveira, Tiago Mendes, Ana Boavida, Ai Nakamura & Margareta Ackermanc. (2019). Co-PoeTryMe: Interactive poetry generation. In *Cognitive Systems Research*, 54, 199–216. <https://doi.org/10.1016/j.cogsys.2018.11.012>
- [17] Hugo Gonalo Oliveira. (2017). O Poeta Artificial 2.0: Increasing meaningfulness in a poetry generation Twitter bot. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)*, 11–20, Santiago de Compostela, Spain. ACL Press. <https://doi.org/10.18653/v1/W17-3902>
- [18] Hugo Gonalo Oliveira. (December, 2015). Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain In *Journal of Artificial General Intelligence*, 6(1), 87–110. <https://doi.org/10.1515/jagi-2015-0005>
- [19] IEEE 730 Standard for Software Quality Assurance Plans. (2014). The Institute of Electrical and Electronics Engineers, Inc. Retrieved from: <https://standards.ieee.org/standard/730-2014.html>.
- [20] Inspired. (November 27, 2020). Google's artificial intelligence algorithm has learned to write poetry, imitating famous poets. Retrieved from: <https://inspired.com.ua/stream/algoritmy-shtuchnogo-intelektu-vid-google-navchyvsya-pysaty-virshinasliduyuchy-vidomyh-poetiv/>
- [21] Istanbul. (2021). JavaScript test coverage made simple. Retrieved from: <https://istanbul.js.org/>.
- [22] Jacobson, R. O. (1975). Linguistics and poetics. Structuralism: pros and cons: Sat. translated verses. Moscow Progress, 193–230. [In Russian].
- [23] Jie Wang, Chengzhi Zhang, Mengying Zhang & Sanhong Deng. (22 Jun 2018). Citation AS: A Tool of Automatic Survey Generation Based on Citation Content. *Journal of Data and Information Science*, 20–37. <https://doi.org/10.2478/jdis-2018-0007>
- [24] Khrolenko, A. T. (2012). *Automated concordance: experience of creation and practice of use*. Retrieved from: <https://cyberleninka.ru/article/n/avtomatizirovannyi-konkordans-opyt-sozdaniya-i-praktika-ispolzovaniya>. [In Russian].
- [25] Kolmogorov, A. N. (2002). Line, stanza and verse as a rhythmic system. *Grinbaum O. N. Materials of the XXXI All-Russian Scientific and Methodological Conference of Teachers and Postgraduates of the Philological Faculty of St. Petersburg State University*, 4(2), 12–28. [In Russian].
- [26] Lotman, M. Yu. (1999). Analysis of the poetic text. The structure of the verse. *Lotman Yu. M. On Poets and Poetry*, 4, 18–253. SPb.: Publishing house of SPb. Retrieved from: <https://www.ruthenia.ru/lotman/papers/apr/>. [In Russian].
- [27] Mandic, D. P., & Chambers, J. A. (2001). *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. Chichester: John Wiley&Sons, 285 p. <https://doi.org/10.1002/047084535X>
- [28] Manurung, H. (2004). An evolutionary algorithm approach to poetry generation. *Doctoral Dissertation for Technical Sciences*. University of Edinburgh. College of Science and En, 367 p.
- [29] Mazur, M. (2015). A Step by Step Backpropagation Example. Matt Mazur. Retrieved from: <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>.
- [30] Michalewicz, Z. (1992). Genetic algorithms + data structures = evolutionary programs. Michalewicz. Charlotte, USA: University of North Carolina, 388 p.
- [31] Miller, S. (2015). Mind: How to Build a Neural Network. Steven Miller. Retrieved from: <http://stevenmiller888.github.io/mind-how-to-build-a-neural-network>
- [32] Mocha: simple, flexible, fun. (2021). Retrieved from: <https://mochajs.org/>.
- [33] Mongo. D. B. (2021). Build faster. Build smarter. Retrieved from: <https://www.mongodb.com/>.
- [34] Neborsina, N. P. (1997). The syntax of poetic speech as a subject of linguopoetic research (based on the material of English and American poetry of the 16th-20th centuries). *Doctoral Dissertation for Philological Sciences* (10.02.04 –

- Germanic Languages). Moscow state university. M. V. Lomonosov, 356 p. [In Russian].
- [35] Oliveira H. G. (2016). PoeTryMe: a versatile platform for poetry generation. Hugo Goncalo Oliveira // CISUC, University of Coimbra, Portugal. Retrieved from: https://eden.dei.uc.pt/~hroliv/pubs/GoncaloOliveira2012_c3gi_CRC.pdf.
- [36] Panchenko, T. V. (2007). *Genetic algorithms*. Astrakhan: Astrakhan University, 86 p. [In Russian].
- [37] Poem Generator. (2021). Masterpiece Generator. Retrieved from: <https://www.poem-generator.org.uk/>.
- [38] Poem. (2019). *Neogranka.ru*. Retrieved from: http://neogranka.ru/generator_stihov.html.
- [39] PoeTryMe. (2014). *University of Coimbra*. Retrieved from: <http://poetryme.dei.uc.pt/>.
- [40] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
- [41] Shevelyova-Garkusha, N. V. (2010). Semantic and functional features of rhythmic-syntactic organization of texts of modern American poetry. *Scientific Bulletin of VN Karazin Kharkiv National University. Ser.: Romano-Germanic philology*, 896(61), 137–143. Retrieved from: <http://eKhSU-IR.kspu.edu/handle/123456789/473>. [In Ukrainian].
- [42] Skulacheva, T. V. (1989). On the question of the interaction of rhythm and syntax in a line of poetry (English and Russian iambic tetrameter). *Izvestia RAN. Literature and Language Series*, 48(2), 156–165. Moscow. [In Russian].
- [43] Slovník.ua. (2005). Ukrainian language and culture portal. *Institute of Linguistics. O. O. Potebny*. Retrieved from: <https://slovník.ua/>. [In Ukrainian].
- [44] Staykova, Kamenka. (15 Jul 2014). Natural Language Generation and Semantic Technologies. *Cybernetics and Information Technologies*, 3–23. <https://doi.org/10.2478/cait-2014-0015>
- [45] Tomashevsky, B. V. (2002). *Literature theory. Poetics: Textbook*. allowance. Moscow: Aspect Press, 334 p. [In Russian].
- [46] Tynyanov, Yu. N. (1993). *Literary fact*. Compiled by O. I. Novikova. Moscow: Higher. shk., 23–121. [In Russian].
- [47] Tynyanov, Yu. N. (1993). The problem of poetic language. *Babylon: Bulletin of Young Literature*, 2(18), 86–90. Moscow: ARGO-RISK. [In Russian].
- [48] Vinogradov, V. V. (1975). From the history of the study of poetics (20s). *Izvestia of the Academy of Sciences of the USSR. Literature and Language Series*, 3, 259–272. Moscow. [In Russian].
- [49] Wang, J., & Hu, S. (2002). Global asymptotic stability and global exponential stability of continuous-time recurrent neural networks. *IEEE Trans. Automatic Control*, 47, 802–807. <https://doi.org/10.1109/TAC.2002.1000277>
- [50] Williams, R. J., & Zipser, D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1, 270–280. <https://doi.org/10.1162/neco.1989.1.2.270>
- [51] Yuanzhi, Ke, & Masafumi, Hagiwara. (03 May 2017). An English Neural Network that Learns Texts, Finds Hidden Knowledge, and Answers Questions. *Journal of Artificial Intelligence and Soft Computing Research*, 229–242. <https://doi.org/10.1515/jaiscr-2017-0016>
- [52] Zakharov, Victor. (21 Dec 2019). Ways of Automatic Identification of Words Belonging to Semantic Field. *Journal of Linguistics*, 234–243. <https://doi.org/10.2478/jazcas-2019-0054>

T. P. Diak, Yu. I. Hrytsiuk

Lviv Polytechnic National University, Lviv, Ukraine

DESIGN OF THE SYSTEM OF AUTOMATED GENERATION OF POETRY WORKS

Features of designing a system of automated generation of poetic works, which opens up new opportunities for artistic speech and show business, especially the preparation of poems and songs have been considered. Quite often lyrics without special content become successful due to the lack of complex plots, as well as due to the unobtrusiveness and ease of perception by listeners. Well-known literature sources and available software products that can generate poetic works by combining different methods and algorithms are analyzed. It has been established that none of them is able to ensure the content and uniqueness of the poetic work at the same time, especially in the Ukrainian language. The existing approaches to the generation of poetic works are analysed, among which the relevant is a method based on templates, generation and testing, evolutionary algorithms and the method based on specific cases. Peculiarities of generating poetical works, first of all rhyming rules, types of strophes, poetic rhythms and sizes have been investigated. An approach to automated generation of poetic works using evolutionary algorithms and a method based on specific cases have been developed. Their combination resembles a sequence of actions for creative personalities when creating poems or writing lyrics.

Peculiarities of neural network organization for automated generation of poetic works have been considered. It is proposed to perform neural network training using the method of inverse propagation and using a genetic algorithm. The principle of operation of algorithms for finding optimal solutions which contain such consecutive stages as initialization, evaluation of solutions, population selection, evolution of solutions, is analysed. Their interaction and various opportunities for neural network learning have been investigated in detail. An algorithm has been developed according to which the software application will analyse the poetic works offered by the user and generate new variants of it on the basis received from the neural network of logically connected words or lines of the verse in the poem. The user can edit both the components of the poem and the generated poetic works, and thus can train the neural network. The specification of requirements to the software application has been developed, the basic requirements to the user interface are defined, and also potential classes of users who will use it are established.

Keywords: computational linguistics; Artificial Intelligence; neural network; genetic algorithm; optimal solution.

Інформація про авторів:

Дяк Тетяна Петрівна, канд. пед. наук, доцент, кафедра прикладної лінгвістики. Email: tetiana.p.diak@lpnu.ua

Грицюк Юрій Іванович, д-р техн. наук, професор кафедри програмного забезпечення.

Email: yurii.i.hrytsiuk@lpnu.ua; <https://orcid.org/0000-0001-8183-3466>

Цитування за ДСТУ: Дяк Т. П., Грицюк Ю. І. Проектування системи автоматизованого генерування віршованих творів.

Український журнал інформаційних технологій. 2021, т. 3, № 2. С. 01–14.

Citation APA: Diak, T. P., & Hrytsiuk, Yu. I. (2020). Design of the system of automated generation of poetry works. *Ukrainian Journal of Information Technology*, 3(2), 01–14. <https://doi.org/10.23939/ujit2021.02.001>