УДК 004.89

<https://doi.org/10.17721/1812-5409.2021/1.13>

Скуржанський О. Г.¹, аспірант,
Марченко О. О.¹, д.ф.-м.н., проф.

O. H. Skurzhanskyi¹, Post-graduate,
A. A. Marchenko¹, Dr. Sci., Prof.

Огляд нейромережевих підходів для задач умовної генерації тексту

Review of neural approaches for conditional text generation

¹Київський національний університет імені Тараса Шевченка, 83000, м. Київ, пр-т. Глушкова 4д,
e-mail: alexskurzh97@gmail.com
rozenkrans17@gmail.com

¹Taras Shevchenko National University of Kyiv, 83000, Kyiv, Glushkova st., 4d,
e-mail: alexskurzh97@gmail.com
rozenkrans17@gmail.com

Стаття присвячена огляду задач умовної генерації тексту: перефразування, виправлення граматичних та орфографічних, спрощення тексту. Розглядаються джерела тренувальних даних, метрики оцінки якості роботи систем та сучасні нейромережеві підходи для розв'язання таких задач. Для кожного завдання розглядається його специфіка та вплив на запропоновані методи. Аналізуються спільні риси задач умовної генерації тексту та їх рішень.

Ключові слова: обробка природної мови, нейронні мережі, машинне навчання, умовна генерація тексту, перефразування, виправлення граматичних помилок, спрощення тексту.

The article is devoted to the review of conditional text generation, one of the most promising fields of natural language processing and artificial intelligence. Specifically, we explore monolingual local sequence transduction tasks: paraphrase generation, grammatical and spelling errors correction, text simplification. To give a better understanding of the considered tasks, we show examples of good rewrites. Then we take a deep look at such key aspects as publicly available datasets with the splits (training, validation, and testing), quality metrics for proper evaluation, and modern solutions based primarily on modern neural networks. For each task, we analyze its main characteristics and how they influence the state-of-the-art models. Eventually, we investigate the most significant shared features for the whole group of tasks in general and for approaches that provide solutions for them.

Key Words: natural language processing, neural networks, machine learning, conditional text generation, paraphrase generation, grammatical error correction, text simplification.

Статтю представив чл.-кор. НАН України, д.ф.-м.н., проф. Анісімов А.В.

Вступ

Генерація тексту є чи не найбільш провідним напрямком обробки природної мови та штучного інтелекту, що найактивніше розвивається на сьогодні. Прорив здійснила нейронна мережа GPT-3 [1], натренована на 45 терабайтах тексту з 18.5 мільярдами параметрів. Вона змогла вирішувати великий спектр задач природної мови, навчаючись на звичайній генерації тексту. Суттєвою частиною цієї області є умовна генерація, коли на вхід подається певна інформація (умова) і текст (вихід) генерується з її

врахуванням. До таких задач належить машинний переклад (умова – текст оригінальною мовою, вихід – текст мовою перекладу), виправлення граматичних та орфографічних помилок (умова – текст з помилками, вихід – текст без помилок), побудова діалогових систем (умова – попередні репліки, вихід – відповідь системи на них).

У цій роботі досліджується поточний стан деяких задач умовної генерації тексту (англійською мовою), а саме перефразування, виправлення граматичних та орфографічних помилок, спрощення тексту. Розглядаються

найефективніші підходи, джерела даних та метрики оцінки якості роботи систем.

Перефразування

Перефразування – переказ тексту іншими словами зі збереженням змісту. Наприклад, питання “How do I talk English fluently?” та “How can I improve my English speaking?” можна вважати парафразами. Особливістю даного напрямку у порівнянні з іншими задачами обробки природної мови є велика кількість робіт, які не використовують розмічені дані, а оперують лише звичайними моно корпусами тексту. Справа в тому, що умова і вихід для цієї задачі є взаємозамінними: якщо з речення $x_1, x_2, \dots, x_m$ ми можемо отримати речення $y_1, y_2, \dots, y_k$ з великою ймовірністю, то й логічно що при умові $y_1, y_2, \dots, y_k$ вихід $x_1, x_2, \dots, x_m$ має мати велику ймовірність. Більше того, кожне речення не повинне мати строго 1 парафраз, а може бути переписано різними способами, що підкреслює ймовірнісну природу задачі.

Для перефразування існує відносно велика кількість джерел даних різної якості та різного рівня (словосполучення, речення, абзаци). Серед них слід виділити:

- Quora Question Pairs (QQP) [2] складається з 404,290 пар питань з сайту Quora, які були помічені модераторами як дублікати і тому є гарантовано парафразами;
- ParaNMT [3] містить більше 50 млн пар англійською, згенерованих нейромережним перекладом паралельних корпусів для задач машинного перекладу. Крім самих пар датасет включає у себе оцінки якості парафразів;
- LanguageNet [4] – набір парафразів отриманий “вирівнюванням” речень твітів, які посилались на один й той самий URL. Він налічує 51,524 пар речень розмічених анотаторами;
- MSCOCO [5] був розроблений для задачі створення субтитрів до зображень. Кожна картинка мала 5 підписів, які з великою ймовірністю є парафразами, що у сумі налічує 117 тисяч пар.

Найпопулярнішими метриками для задачі перефразування є BLEU [6] та ROUGE [7]. BLEU обчислюється наступним чином: для кожного $n \in [1, 2, 3, 4]$ рахуємо частку n -грам у реченні-кандидаті, які потрапляють у розмічене (правильне) речення і усереднюємо отримані числа. Далі множимо експоненту середнього на штраф за коротку довжину речення.

$$p_n = \frac{\sum \text{count}_{\text{clip}}(n\text{-gram})}{\sum \text{count}(n\text{-gram})}$$

$$B = \begin{cases} \exp(1 - |ref|/|hyp|), \text{ якщо } |ref| > |hyp| \\ 1, \text{ інакше} \end{cases}$$

$$BLEU = B \cdot \exp\left[\frac{1}{N} \sum p_n\right]$$

Таким чином, BLEU намагається порахувати відсоток “правильних” n -грам у реченні, при цьому щоб отримане речення не було коротше за “золоте”. ROUGE – набір метрик, які враховують повноту: ROUGE- n – рахує F_1 міру для n -грам аналогічно як точність рахується у BLEU, а ROUGE-L рахує найдовшу підпоследовність tokenів з речення-кандидата у розміченому реченні.

Проаналізувавши літературу по перефразуванню можна прийти до наступного висновку: найкращою системою для поточної задачі є LBOW-Topk з роботи Paraphrase Generation with Latent Bag of Words [8]. Модель – енкодер-декодер LSTM з модифікаціями для врахування специфіки задачі перефразування.

У класичному seq2seq підході модель спочатку кодує вхідну последовність $x = x_1, x_2, \dots, x_m$ у последовність h латентного простору, а потім декодує її у цільову последовність y .

Енкодер enc_ϕ та декодер dec_θ – нейронні мережі, а функція втрат рахується наступним чином:

$$\begin{aligned} h &= enc_\phi(x) \\ p(y|x) &= dec_\theta(h) \\ L_{S2S} &= E_{(x^*, y^*) \sim p} [-\log p_\theta(y^* | x^*)], \end{aligned}$$

де P^* – справжній розподіл даних.

Запропонована ж модель використовує слова з вхідного речення для передбачення їх “сусідів” в мішку слів з цільового речення. З передбачених “сусідів” береться випадковим чином підмножина слів, яка й використовується для передбачення цільового речення. Робиться це наступним чином.

Нехай маємо словник розміру V , тоді мішок слів z розміру k – це вектор з R^V , де k – кількість одиниць, а всі інші значення нулі. Припускається, що z отримується шляхом семплінгу з базового категоріально розподілу $p(\tilde{z}|x)$ k разів без замінів. Для кожного слова x_i з вхідної последовності моделюється “сусід” $z_{ij} \in R^V$:

$$p(z_{ij} | x_i) = \text{Categorical}(\phi_{ij}(x_i)),$$

де ϕ_{ij} – нейронна мережа. На практиці застосовується лінійний шар з softmax на виході h енкодера enc_ϕ . Припускається, що кожен вихідний token x_i має максимальну кількість

сусідів l . Далі ймовірності всіх “сусідів” переміщуються усередненням:

$$\tilde{z} \sim p_{\phi}(\tilde{z} | x) = \frac{1}{ml} \sum_{i,j} p(z_{ij} | x)$$

Як було згадано вище, мішок слів z семплиться k разів без заміни з розподілу $p(\tilde{z} | x)$. Тоді z використовується для контролюваного декодування вихідної послідовності у:

$$z \sim p_{\phi}(\tilde{z} | x)$$

$$y \sim p_{\theta}(y | x, z) = \text{dec}_{\theta}(x, z)$$

Тоді фінальна функція втрат враховує як коректність передбачення мішку слів, так і декодування:

$$L_{S2S'} = E_{(x^*, y^*) \sim p, z \sim p_{\phi}(\tilde{z} | x)} [-\log p_{\theta}(y^* | x^*, z)]$$

$$L_{BOW} = E_{z \sim p^*} [-\log p_{\phi}(z^* | x)]$$

$$L = L_{S2S'} + L_{BOW},$$

де z^* представляє мішок слів цільової послідовності.

Сам по собі семплінг z є недиференційованим. Щоб мати змогу рахувати градієнти зворотного поширення помилки використовується параметрична оцінка градієнтів через gumbel-softmax. Нехай ймовірність $\tilde{z}$ дорівнює $p(\tilde{z} = i | x) = \pi_i, i \in 1, \dots, V$. Збурені ваги і ймовірності отримуються наступним чином:

$$a_i = \log \pi_i + g_i$$

$$g_i \sim \text{Gumbel}(0, 1)$$

Вибір k найбільших вагів з $x = x_1, x_2, \dots, x_V$ дасть той самий результат, що й семплінг k разів без заміни.

Виправлення граматичних та орфографічних помилок

Назва задачі говорить сама за себе: на вхід (як умова) подається речення, в якому потенційно можуть бути помилки, на вихід система має видати виправлене речення, тобто без помилок. Характерною особливістю напрямку є те, що вихідне речення не сильно відрізняється від вхідного – відносно мала відстань Левенштейна по словам. Через це багато робіт у даній області не переписують речення повністю, а лише пропонують виправлення. Таким чином, підходи з маркування послідовностей працюють на рівні з seq2seq.

Для задачі виправлення граматичних та орфографічних помилок існує доволі велика кількість розмічених даних, більшість з яких отримані з курсів вивчення англійської. Серед них:

- National University of Singapore Corpus of Learner English (NUCLE) [9] містить 1,400 есе написаних студентами та проанотованих професійними викладачами англійської. Всього налічує 56,958 речень, кожне з яких проанотоване 2 спеціалістами;

- Lang-8 Learner Corpora [10] є найбільшим серед відомих відкритих корпусів. Містить майже мільйон пар речень англійською з сайту Lang-8, де користувачі виправляють граматику один одного;

- First Certificate in English (FCE) [11] corpus складається 1,244 виправлених відповідей до екзаменаційних питань FCE;

- W&I+LOCNESS [12] містить 34,304 пар речень, які були написані студентами та розмічені професійними аотаторами з Write & Improve. Крім самих пар міститься інформація про те, який рівень англійської здавав студент.

Оскільки існує консенсус як має виглядати граматично правильне речення англійської, а виправлені речення не сильно відрізняються від оригінальних, оцінювання переписувань систем є більш простим та зрозумілим у порівнянні з іншими задачами природної мови. Зокрема застосовується MaxMatch (M2) Scorer [13] та ERRANT [14]. Обидві метрики є нічим іншим як $F_{0.5}$ мірою кількості правильних виправлень. Різниця між ними тільки у алгоритмі підрахунку виправлень.

$$F_{0.5} = \frac{1.25 \cdot \text{precision} \cdot \text{recall}}{0.25 \cdot \text{precision} + \text{recall}}$$

Найкращою системою для виправлення граматичних і орфографічних помилок є GECToR з роботи GECToR – Grammatical Error Correction: Tag, Not Rewrite [15]. Модель, замість того щоб передбачати правильне речення, передбачає виправлення, які потрібно зробити. З точки зору архітектури – це трансформер XLNet [16], дотренований на передбаченні граматичних та орфографічних виправлень. Нейронна мережа намагається вирішити одразу 2 задачі: передбачити чи потрібно робити виправлення для поточного токена x_i послідовності $x = x_1, x_2, \dots, x_m$ та яке саме виправлення потрібно зробити. Для цього отримане після енкодера представлення h_i подається одразу на 2 лінійні шари:

$$h = \text{enc}_{\phi}(x)$$

$$p_l = W_l h + b_l$$

$$p_d = W_d h + b_d.$$

Далі для обох передбачень рахується функція втрат та сумується:

$$L = L_{\text{labeling}} + L_{\text{detection}}$$

Існують різні типи виправлень, які передбачає модель:

- KEEP – не чіпати поточний токен;
- DELETE – видалити поточний токен;
- APPEND_ w_i – вставити слово w_i після поточного токена;
- REPLACE_ w_i – замінити поточний токен на слово w_i ;
- CASE – змінити регістр поточного токена;
- MERGE – “з’єднати” поточний токен з наступним;
- SPLIT – розбити поточний токен на два;
- NOUN NUMBER – змінити множину поточного іменника;
- VERB FORM – змінити форму поточного дієслова

Оскільки дана модель є неавторегресивною та має лише енкодер (декодер відсутній), вона є відносно швидкою (приблизно в 2-3 рази за авторегресивні моделі).

Спрощення тексту

Спрощення тексту — переписування тексту таким чином, що граматику та структуру значно спрощуються, а основне значення та інформація залишаються незмінними. Наприклад, для речення “Both men and women when attending a mosque must adhere to these guidelines.” спрощеним буде “Both men and women when going a mosque must follow these rules.”

В порівнянні з попередніми задачами джерел даних для задачі спрощення тексту помітно менше. Серед них:

- WikiLarge [17] складається з 296 тисяч пар речень (складних і відповідних простих), створених напівавтоматичним чином зі змін у Вікіпедії;
- Newsela [18] містить 1,130 статей новин, переписаних анотаторами для дітей різних класів з різним “рівнем” спрощення. Налічує 95 тисяч пар речень та є більш якісним у порівнянні з WikiLarge.

Як основні метрики для оцінки якості систем спрощення тексту використовуються SARI [19] та FKGL [20]. SARI рахується як середнє арифметичне між F_1 мірами для n -грам трьох операцій add (додавання нових токенів), keep (для токенів, що залишилися у спрощених варіантах переписувань систем та розмічених речення) та delete (видалення токенів з оригінального речення).

$$ope \in [add, keep, del]$$

$$f_{ope}(n) = \frac{2 \cdot p_{ope}(n) \cdot r_{ope}(n)}{p_{ope}(n) + r_{ope}(n)}$$

$$F_{ope} = \frac{1}{k} \sum_{n=1, \dots, k} f_{ope}(n)$$

$$SARI = \frac{F_{add} + F_{keep} + F_{del}}{3}$$

FKGL – міра читабельності речення; лінійна комбінація кількості слів на речення та кількості складів на слово:

$$FKGL = 0.39 \frac{\# words}{\# sentences} + 11.8 \frac{\# syllables}{\# words} - 15.59$$

Метрика не використовує правильні (розмічені) речення, а тому має застосовуватися у комбінації з SARI.

Згідно вищезазначених метрик якості, найкращою системою на сьогоднішній день є нейронна мережа з роботи Multilingual Unsupervised Sentence Simplification [21]. Це seq2seq модель архітектури transformer - BART [22].

У роботі значна увага приділяється контрольованій генерації тексту. Для цього під час навчання на вхід, крім оригінальних (складних) речень подаються також спеціальні токени, які відповідають певним атрибутам спрощення:

- NbChars – співвідношення літер у оригінальному і спрощеному реченні. Відповідає за рівень “компресії” переписаного;
- LevSim – нормалізована схожість Левенштейна між реченнями на рівні літер. Тобто на скільки відрізняється переписане речення від оригінального;
- WordRank вимірює те, на скільки слова з отриманого речення є вживаними у порівнянні з вхідним;
- DepTreeDepth відповідає глибині Dependency tree для речень. Порівнює синтаксичну складність.

Далі під час генерації тексту подаються ці 4 спеціальні токени в залежності від того, яким ми хочемо бачити переписане речення. Наприклад, щоб отримати максимально коротке речення можна подати токен, який відповідає NbChars = 0.1.

Порівняльний аналіз

Слід зазначити, що вищезгадані задачі мають спільну рису – в усіх задачах переписування тексту відбувається зі збереженням змісту. Тобто якщо “форма” змінюється, то сенс залишається тим самим. Підходи до вирішення задач теж мають ряд спільного.

Насамперед це модель трансформер. З моменту її представлення у роботі Attention is all

you need [23], трансформери стали де-факто стандартом для state of the art підходів у області обробки природної мови. Суттєву роль у цьому зіграло так зване попереднє тренування моделей на більш загальних текстових задачах. Зазвичай для таких задач не потрібні розмічені дані – достатньо великих моно корпусів для анотацій. Наприклад, у найпопулярнішому варіанті Masked Language Modelling, 15% випадкових tokenів маскуються і моделі слід відгадати справжні.

Другим спільним моментом усіх підходів є те, що створюються архітектурні особливості під кожну задачу. Так у задачі перефразування для врахування стохастичності виходу (для речення не існує строго одного парафраза) використовується Latent Bag of Words, який допомагає випадковим чином вибрати токени майбутнього речення та, застосовуючи їх, згенерувати текст. Для врахування того факту, що речення суттєво не змінюється у виправленні граматичних помилок передбачаються виправлення замість генерації всього тексту. У спрощенні тексту в модель закладаються її атрибути: наскільки текст має бути коротший, простіший (з лексичної та синтаксичної точки зору) та як сильно відрізнятись від вхідного.

Третя риса – архітектура seq2seq, що є універсальною на сьогоднішній день. Покращена cross-attention з трансформеру, seq2seq дуже добре передає контекст отриманий з енкодера у декодер. Звісно це призводить до авторегресивної генерації, що у свою чергу сповільнює модель, але одночасно дає хорошу якість.

Список використаних джерел

1. Brown T. B. et al. Language Models are Few-Shot Learners – Режим доступу до статті: <https://arxiv.org/abs/2005.14165>
2. Quora Duplicate Questions | Kaggle [Електронний ресурс] – Режим доступу: <https://www.kaggle.com/aymenmouelhi/quora-duplicate-questions>
3. Wieting J., Gimpel K. PARANMT-50M: Pushing the Limits of Paraphrastic Sentence Embeddings with Millions of Machine Translations – Режим доступу до статті: <https://www.aclweb.org/anthology/P18-1042.pdf>
4. Lan W., Qiu S., He H., Xu W. A Continuously Growing Dataset of Sentential Paraphrases – Режим доступу до статті: <https://www.aclweb.org/anthology/D17-1126.pdf>
5. Lin T. et al. Microsoft COCO: Common Objects in Context – Режим доступу до статті: <https://arxiv.org/pdf/1405.0312.pdf>
6. Papineni K., Roukos S., Ward T., Zhu W. BLEU: a Method for Automatic Evaluation of Machine Translation – Режим доступу до статті: <https://www.aclweb.org/anthology/P02-1040.pdf>
7. Lin C. ROUGE: A Package for Automatic Evaluation of Summaries – Режим доступу до статті: <https://www.aclweb.org/anthology/W04-1013.pdf>
8. Fu Y., Feng Y. Paraphrase Generation with Latent Bag of Words Summaries – Режим доступу до статті: <https://arxiv.org/pdf/2001.01941.pdf>
9. Dahlmeier D., Ng H. T., Wu S. M. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English Summaries – Режим доступу до статті: <https://www.aclweb.org/anthology/W13-1703.pdf>
10. Tajiri T., Komachi M., Matsumoto Y. Tense and Aspect Error Correction for ESL Learners Using Global Context – Режим доступу до статті: <https://www.aclweb.org/anthology/P12-2039.pdf>
11. Yannakoudakis H., Briscoe T., Medlock B. A New Dataset and Method for Automatically Grading ESOL Texts Context – Режим доступу до статті: <https://www.aclweb.org/anthology/P11-1019.pdf>
12. Bryant C., Felice M., Andersen Ø.E., Briscoe T. The BEA-2019 Shared Task on Grammatical Error Correction – Режим доступу до статті: <https://www.aclweb.org/anthology/W19-4406.pdf>
13. Dahlemeier D., Ng H.W. Better Evaluation for Grammatical Error Correction – Режим доступу до статті: <https://www.aclweb.org/anthology/N12-1067.pdf>
14. Bryant C., Felice M., Briscoe T. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction – Режим доступу до статті: <https://www.aclweb.org/anthology/P17-1074.pdf>
15. Omelianchuk K., Atrasevych V., Chernodub A., Skurzhanyskiy O. GECToR – Grammatical Error Correction: Tag, Not Rewrite – Режим доступу до статті: <https://www.aclweb.org/anthology/2020.bea-1.16.pdf>
16. Yang Z. et al. XLNet: Generalized Autoregressive Pretraining for Language

- Understanding Rewrite – Режим доступу до статті: <https://arxiv.org/pdf/1906.08237.pdf>
17. Zhang X., Lapata M. Sentence Simplification with Deep Reinforcement Learning – Режим доступу до статті: <https://www.aclweb.org/anthology/D17-1062.pdf>
 18. Xu W., Callison C., Napoles C. Problems in Current Text Simplification Research: New Data Can Help – Режим доступу до статті: <https://www.aclweb.org/anthology/Q15-1021.pdf>
 19. Xu W. et al. Optimizing Statistical Machine Translation for Text Simplification – Режим доступу до статті: <https://www.aclweb.org/anthology/Q16-1029.pdf>
 20. Kincaid J.P. et al. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel
 21. Martin L. et al. MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases – Режим доступу до статті: <https://arxiv.org/pdf/2005.00352.pdf>
 22. Lewis M. et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension – Режим доступу до статті: <https://arxiv.org/pdf/1910.13461.pdf>
 23. Vaswani A. et al. Attention Is All You Need – Режим доступу до статті: <https://arxiv.org/pdf/1706.03762.pdf>
- ### References
1. BROWN, T.B. et al. (2020) Language Models are Few-Shot Learners. In *NeurIPS 2020*
 2. KAGGLE. (2017) *Quora Duplicate Questions* [Online] – Available from: <https://www.kaggle.com/aymenmouelhi/quora-duplicate-questions> [Accessed: 19th June 2012].
 3. WIETING, J. and GIMPEL, K. (2017) PARANMT-50M: Pushing the Limits of Paraphrastic Embeddings with Millions of Machine Translations. In *ACL 2017*
 4. LAN, W. et al. (2017) A Continuously Growing Dataset of Sentential Paraphrases. In *EMNLP 2017*
 5. LIN, T. et al. (2014) Microsoft COCO: Common Objects in Context. In *ECCV 2014*
 6. PAPINENI, K. et al. (2002) Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL 2002*
 7. LIN, C. (2004) ROUGE: A Package for Automatic Evaluation for Summary. In *ACL 2004*
 8. FU, Y. and FENG, Y. (2019) Paraphrase Generation with Latent Bag of Words. In *NeurIPS 2019*
 9. DAHLMEIER, D. et al. (2013) Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *BEA 2013*
 10. TAJIRI, T. et al. (2012) Tense and Aspect Error Correction for ESL Learners Using Global Context. In *ACL 2012*
 11. YANNAKOUDAKIS, H. et al. (2011) A New Dataset and Method for Automatically Grading ESOL Texts. In *ACL 2011*
 12. BRYANT, C. et al. (2019) The BEA-2019 Shared Task on Grammatical Error Correction. In *ACL 2019*
 13. DAHLMEIER, D. and NH, T. H. (2012) Better Evaluation for Grammatical Error Correction. In *NAACL 2012*
 14. BRYANT, C. et al. (2017) Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *ACL 2017*
 15. OMELIANCHUK, K. et al. (2019) GECToR – Grammatical Error Correction: Tag, Not Rewrite. In *BEA 2019*
 16. YANG, Z. et al. (2019) XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NeurIPS 2019*
 17. ZHANG, X. and LAPATA, M. (2017) Sentence Simplification with Deep Reinforcement Learning. In *EMNLP 2017*
 18. XU, W. et al. (2015) Problems in Current Text Simplification Research: New Data Can Help. In *TACL 2015*
 19. XU, W. et al. (2016) Optimizing Statistical Machine Translation for Text Simplification. In *TACL 2016*
 20. KINCAID, J. P. et al. (1975) *Derivation Of New Readability Formulas*. Institute for Simulation and Training, 56
 21. MARTIN, L. et al. (2020) Multilingual Unsupervised Sentence Simplification
 22. LEWIS, M. et al. (2019) BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL 2020*
 23. VASWANI, A. (2017) Attention is all you need. In *NIPS 2017*

Надійшла до редколегії 29.01.21