

УДК 004.89

<https://doi.org/10.17721/1812-5409.2023/2.40>

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

O. H. Skurzhanyskyi¹, Post-graduate,
A. A. Marchenko¹, Dr. Sci., Prof.

Нейромережеві підходи для задач письмового асистента

Neural approaches for writing assistant tasks

¹Київський національний університет імені
Тараса Шевченка, 83000, м. Київ, пр-т. Глушкова
4д,
e-mail: alexskurz97@gmail.com
rozenkrans17@gmail.com

¹Taras Shevchenko National University of Kyiv,
83000, Kyiv, Glushkova st., 4d,
e-mail: alexskurz97@gmail.com
rozenkrans17@gmail.com

Стаття присвячена дослідженню та аналізу задач для побудови письмового асистенту: виправлення граматичних та орфографічних помилок, спрощення тексту та перефразування. Розглядаються розмічені набори даних, метрики визначення якості роботи систем та провідні практики вирішення для розв'язання таких задач з використанням нейронних мереж. Для кожної задачі розглядається його специфіка та вплив на запропоновані методи. Аналізуються спільні риси підходів до вирішення задач письмового асистента та їх рішень.

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

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

Keywords: natural language processing, neural networks, machine learning, writing assistant, paraphrasing, grammar correction, text simplification.

Статтю представив д.ф.-м.н., проф. Терещенко В. М.

Вступ

Текстова генерація стала сьогодні одним з найважливіших напрямків як у сфері обробки природної мови так і штучного інтелекту в цілому. Велика мовна модель ChatGPT [1], яка була натренована на суттєвій частині інтернету даних та має сотні мільярди параметрів, зробила значний прогрес у цій області, виконуючи широкий спектр завдань на основі генерації тексту. Важливою частиною цього напрямку є умовна генерація тексту, де не просто передбачається текст

природної мови, а й враховується інший текст як вхідна інформація або умова. Така постановка задачі включає у себе величезний спектр завдань, включно з діалоговими системами, виправленням помилок у кодї, сумаризацією, тощо. В даному дослідженні розглядається актуальний стан деяких завдань умовної генерації пов'язаних з письмовим асистентом: виправлення граматичних та орфографічних помилок, спрощення контенту та перефразування. Аналізуються провідні практики вирішення задач, джерела даних та критерії якості для таких систем.

Виправлення граматичних та орфографічних помилок

Задача виправлення помилок є базовою для письмових асистентів. Постановка наступна: на вхід (як умова) подається речення, в якому потенційно можуть бути помилки, на вихід система має згенерувати виправлене речення, тобто без помилок. Характерною особливістю напрямку є те, що різниця між вхідними та вихідними реченнями відносно мала з точки зору таких метрик як відстань Левенштейна по токенам. Через це суттєва частина робіт у даній області не переписують речення з нуля, а лише пропонують виправлення до нього. Таким чином, моделі, що роблять передбачення виправлень для кожного токена (sequence labeling), працюють на рівні з повним переписуванням (seq2seq).

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

- W&I+LOCNESS [2] містить 34,304 пар речень, які були написані студентами та розмічені професійними анотаторами з Write & Improve. Крім самих пар міститься інформація про те, який рівень англійської здавав студент. Вважається найякіснішим набором даних; має тренувальну та тестову частину;

- National University of Singapore Corpus of Learner English (NUCLE) [3] містить 1,400 есе написаних студентами та проанотованих професійними викладачами англійської. Всього налічує 56,958 речень, кожне з яких проанотоване двома спеціалістами; має тренувальну та тестову частину;

- Lang-8 Learner Corpora [4] є найбільшим з точки зору кількості прикладів корпусом. Містить майже мільйон пар речень англійською з сайту Lang-8, де користувачі виправляють граматику один одного. Вважається найгіршим з точки зору якості;

- First Certificate in English (FCE) [5] corpus складається 1,244 виправлених відповідей до екзаменаційних питань FCE.

MaxMatch (M2) scorer [6] використовується найчастіше для оцінки якості задач виправлення граматичних помилок (GEC), куди часто включають і орфографічні. Оскільки існує консенсус як має виглядати граматично правильне речення англійської, а виправлені речення не сильно відрізняються від оригінальних,

оцінювання переписувань таких систем є більш простим та зрозумілим у порівнянні з іншими задачами природної мови.

M2 розраховується на основі знаходження найбільшої послідовності співпадаючих tokenів між вихідним (згенерованим) текстом системи та декількома допустимими варіантами, які вважаються "правильними" (зазвичай це текст, оцінений людськими експертами). Таким чином рахується точність (precision) та повнота (recall), а далі застосовується $F_{0.5}$, який пріоритизує перше над другим приблизно у 2 рази.

Також широкого розповсюдження набула метрика ERRANT (ERRor ANnotation Toolkit) [7], який є спеціалізованим інструментом для порівняння та анотації мовних помилок в англійському тексті. Відмінністю ERRANT від M2 є те, що вона фокусується на типах помилок та їх класифікації, замість просто врахування кількості. Це робить її корисною для більш змістовного оцінювання та аналізу систем автоматичного виправлення помилок.

Основна ідея ERRANT полягає в тому, щоб розбити зміни тексту на атомарні операції, такі як вставка, видалення або заміна слова, і потім класифікувати ці зміни за типами помилок, наприклад, орфографічні, граматичні або стилістичні. Це дозволяє отримати докладний аналіз роботи системи та виявити, на яких типах помилок вона працює найкраще чи найгірше.

Згідно з метрикою ERRANT, однією з найкращих архітектур для виправлення граматичних та орфографічних помилок була представлена у роботі Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data [8]. Це модифікація трансформера з використанням так званого механізму копіювання.

На кожному кроці генерації декодера, окрім вибору tokenів з фіксованого словника, нейронна мережа «збагачена» механізмом копіювання має можливість вибирати токени з вхідного речення. Таким чином фінальний розподіл P_t це комбінація розподілу генерації P_t^{gen} та розподілу копіювання P_t^{copy} . В результаті маємо розширення вхідного словника токенами з вхідного тексту. Баланс між копіюванням та генерацією досягається завдяки змінній $\alpha_t^{copy} \in [0,1]$ на кожному кроці часу t :

$$p_t(w) = (1 - \alpha_t^{copy}) * p_t^{gen}(w) + \alpha_t^{copy} * p_t^{copy}(w)$$

Якщо обрахунок P_t^{gen} відбувається стандартним чином, то P_t^{copy} рахується за допомогою нового розподілу уваги між поточним прихованим станом декодера h_t^{trg} та прихованими станами енкодера H^{src} . Увага копіювання рахується

схожим чином, як увага між енкодером та декодером:

$$q_t, K, V = h_t^{trg} W_q^T, H^{src} W_k^T, H^{src} W_v^T$$

$$A_t = q_t^T K$$

$$P_t^{copy}(w) = softmax(A_t)$$

Далі використовується нормалізований розподіл уваги та приховані стани копіювання для обрахунку балануючої змінної α_t^{copy} :

$$\alpha_t^{copy} = \sigma(W^T \sum(A_t^T V)).$$

У роботі Synthetic Data Generation for Grammatical Error Correction with Tagged Corruption Models [9] розглядається інший напрямок для покращення систем виправлення граматичних та орфографічних помилок — генерація синтетичних даних для подальшого претренування.

Якщо у більшості робіт у тексти додаються помилки завдяки правилам, які спрацьовують випадково, то тут будується окрема нейронна мережа для створення помилок у тексті. Відбувається це наступним чином. Спочатку береться звичайний тренувальний датасет для задачі виправлення помилок, він розмічається завдяки інструменту ERRANT на типи помилок. Далі модель вчиться генерувати текст з помилками з правильного тексту. Якщо зазвичай через ймовірнісну природу генерації помилок, моделі важко зорієнтуватися, яку саме внести опіску та у якому місці, то у даному випадку тип помилки подається на вхід моделі як текстовий префікс.

Таким чином ми краще контролюємо нашу модель для внесення помилок та можемо зробити розподіл помилок відповідним, наприклад, тестовій частині датасету. Саме це і роблять автори повторюючи розподіл валідаційної частини W&I+LOCNESS. Згенеровані дані обіцяють викласти у відкритий доступ.

Автори пропонують наступні кроки для отримання моделей найкращої якості:

1. Претренування моделі на отриманому синтетичному датасеті разом з помилками з Вікіпедії
2. Тренування на датасеті Lang-8
3. Остаточне дотренування на комбінації FCE та W&I+LOCNESS

Так ми рухаємося від найбільш зашумлених джерел даних до найбільш якісних.

Спрощення тексту

Спрощення тексту - це переформулювання тексту зі спрощенням граматики та структури,

зберігаючи при цьому основний сенс та вміст. До прикладу, речення “All men and women when attending a mosque must adhere to these guidelines.” можна спростити до “All men and women when going to a mosque must follow these rules.”

Якісних публічних джерел для задачі спрощення тексту не так багато. Серед них найбільш використовувані:

- Newsela [10] має 1,130 статей новин, переписаних анотаторами для дітей різних класів з різним “рівнем” спрощення. Набір даних у сумі налічує 95 тисяч пар речень та є більш якісним у порівнянні з іншими;

- WikiLarge [11] містить з 296 тисяч пар речень (складних і відповідних простих), створених напівавтоматичним чином з історії змін у Вікіпедії.

Як основні метрики для оцінки якості систем спрощення тексту використовуються SARI [12] та FKGL [13]. Метрика SARI (System output Against Reference and Input) розроблена спеціально для оцінки якості текстових перетворень, таких як спрощення тексту. Вона враховує як аспекти точності, так і повноти, та оцінює три ключові компоненти: видалені слова (delete), додані слова (add) і збережені слова (keep). Формально, SARI розраховується як середнє арифметичне цих трьох компонентів:

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

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

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

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

FKGL у свою чергу ще у 1975 для тесту на читабельність. Це 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 (анотовані) речення, а тому часто застосовується у комбінації з SARI.

У роботі Integrating Transformer and Paraphrase Rules for Sentence Simplification [14] автори намагаються додати зовнішні знання до нейронної мережі архітектури трансформер. Джерелом у даному випадку виступає Paraphrase Database for Simplification (PPDB) [15] — відкрита база даних для спрощення слів/словосполучень. Наприклад, пара «ключ» → «значення»: recipient → winner.

Перш за все, використовується спеціальна функція втрат, яка зменшує ймовірність складних слів у реченні (тих, що потрапили як «ключі» до

бази даних) та збільшує ймовірність простих слів (тих, що потрапили як «значення»).

По-друге, правила з PPDB враховуються на кожному кроці генерації. Спочатку рахується вектор контексту як середнє зважене виходів енкодера, де ваги це скалярний добуток з поточним прихованим вектором декодера. Далі цей вектор контексту проходиться по всім представленням правил з PPDB для слова, яке генерується на поточному кроці. Представлення слів, запропонованими правилами зважуються і додаються до поточного виходу енкодера. Таким чином нейронна мережа враховує правила з бази даних.

Ще однією знаковою роботою для задачі спрощення тексту є Text Simplification by Tagging [16]. У ній автори використовують вже знайому нам архітектуру GECToR [17], але для цієї задачі.

Ця модель, замість того щоб передбачати правильне речення, передбачає виправлення, які потрібно зробити. З точки зору архітектури – це трансформер енкодер XLNet [18], дотренований на передбаченні граматичних та орфографічних виправлень. Таким чином модель передбачає виправлення паралельно, у неавторегресивній манері, на відміну від класичних підходів.

Існують різні типи виправлень, які генерує модель. Серед них

- 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 .

Сама по собі така модель показує недостатні результати, тому автори намагаються використовувати різні прийоми для її покращення. Розглянемо їх далі.

BackTranslation: для збільшення розміру тренувальних даних вхідні речення з датасету проганяються через моделі машинного перекладу туди назад. Наприклад, спочатку з англійської на німецьку, а далі назад з німецької на англійську. Таким чином отримується парафраз для вхідного речення, який можна використати при тренуванні основної моделі.

Ensemble Generation: на основі поточних найкращих моделей будується ансамбль, який генерує більш якісні дані, але значно повільніше. Далі ансамблем з вхідних речень тренувального датасету генеруються спрощення. Такі пари теж використовуються під час тренування фінальної моделі.

Ці кроки дозволяють помітно покращити якість моделі. Фінальна модель є другою найкращою на поточний момент згідно метрики SARI.

Перефразування

Перефразування — це переклад того самого тексту різними словами без втрати сенсу самого тексту. Іншими словами, ми передаємо зміст тексту своїми словами. Найчастіше, перефразований текст може передавати своє значення краще, ніж оригінальні слова, і фактично, роблячи це, ми створюємо копію тексту за значенням, але яка відрізняється від оригіналу з точки зору використаних слів та їх порядку. Наприклад, питання “How do I talk English fluently?” та “How can I improve my English speaking?” можна вважати парафразами. Ця задача відрізняється від інших завдань обробки природної мови тим, що є велика кількість робіт, які не використовують розмічені дані, а оперують лише звичайними корпусами тексту без анотацій. Це стає можливим завдяки 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 має мати строго 1 парафраз, а може бути переписано різними способами, що підкреслює ймовірнісну природу задачі.

Для перефразування існує доволі велика кількість джерел даних різної якості. Серед них варто виділити:

- ParaNMT [19] містить більше 50 млн пар англійською, згенерованих моделлю машинного навчання перекладом паралельних корпусів. Крім самих пар датасет включає у себе оцінки якості парафразів;
- Quora Question Pairs (QQP) [20] складається з 404 тисяч пар питань з соціального сервісу Quora, на якому користувачі можуть ставити відкриті для аудиторії питання та отримувати відповіді. Ці питання були помічені модераторами як дублікати і тому є гарантовано парафразами;
- MSCOCO [21] був оригінально розроблений для задачі створення субтитрів до зображень. Кожне зображення мало 5 описів, що у сумі налічує 117 тисяч пар. З великою ймовірністю описи для того самого зображення є парафразами.

Найпопулярнішими метриками для задачі перефразування є BLEU [22] та METEOR [23]. BLEU (BiLingual Evaluation Understudy) — це одна з найпоширеніших метрик для оцінки якості

машинного перекладу та інших завдань генерації тексту. Вона була введена у 2002 році і з того часу стала стандартом в оцінці машинного перекладу. Основна ідея BLEU полягає в порівнянні n-грамів у вихідному тексті (згенерованому тексті) з n-грамами в розміченому тексті (оригінальний текст або "золотий стандарт") з точки зору точності співпадінь. Точний підрахунок метрики відбувається наступним чином.

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

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

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

Метрика ROUGE (Recall-Oriented Understudy for Gisting Evaluation) є іншою популярною метрикою для оцінки якості генерації тексту, зокрема, у задачах текстового стиснення та анотування. Вона фокусується на вимірюванні відтворюваності (Recall) між згенерованим текстом і анотованим (золотим стандартом).

Основні форми ROUGE включають у себе три компоненти для підрахунку метрики:

- ROUGE-N: Співставлення n-грамів у згенерованому та посилальному текстах.
- ROUGE-L: Найдовша спільна підпоследовність (LCS) між двома текстами.
- ROUGE-S: Відстань між послідовними словами в тексті.

Хорошим прикладом побудови системи перефразування без використання відповідних тренувальних даних є робота Zero-Shot Paraphrase Generation with Multilingual Language Models [24]. Автори стверджують, що стандартний підхід генерації парафразів через подвійний переклад є не надійним та часто може змінювати зміст речень. Натомість пропонують використовувати датасети машинного перекладу для цієї задачі на мовних моделях архітектури трансформер (де тільки декодер).

Спочатку тренується стандартна мовна модель одразу на декількох мовах, для цього на вхід подаються службові токени – початковий і кінцевий, токен, що відповідає мові, та сам текст. Наприклад, *<bos> <en> cat sat on the mat <eos>* або *<bos> <fr> j'aime le football <eos>*. Таким чином, модель вивчає як будуються речення різними мовами, а мовний токен дає можливість контролювати якою мовою ми генеруємо текст.

Далі використовується датасети з машинного навчання. На вхід подаються пари речень, які є перекладами один одного. Знову застосовуються мовні токени, а також токен розділення речень.

Наприклад, *<bos> <en> cat sat on the mat <dim> <delim> <fr> chat assis sur le tapis <eos>*.

В результаті, щоб отримати парафраз як префікс другого речення подається мовний токен тієї ж самої мови. Наприклад, на вхід *<bos> <en> cat sat on the mat <dim> <delim> <en>*, а на вихід *<bos> <en> cat sat on the mat <dim> <delim> <en> the cat sitting on the carpet <eos>*.

Також для покращення результату використовують додатково представлення мов, аби остаточно змусити модель лишатися на одній мові, зашумлення вхідних речень, аби модель лишалась інваріантною до помилок.

Іншою цікавою роботою для задачі перефразування є Learning Semantic Sentence Embeddings using Pair-wise Discriminator [25]. У ній автори тренують не тільки енкодер та декодер, а й дискримінатор.

Енкодер та декодер навчаються зі стандартною функцією втрат – кросс ентропією. Далі дискримінатор, який має спільні ваги з енкодером, будує векторні представлення для згенерованого енкодером-декодером парафраза та для цільового речення. Автори пропонують максимізувати скалярний добуток відповідних представлень передбачених й розмічених речень та мінімізувати його для всіх інших речень у батчі.

В результаті ми отримуємо поєднання локальної функції втрат (для кожного токена згенерованого декодером) та глобальної (схожість векторних представлень). Таким чином, ми намагаємося також врахувати загальний зміст речення. Після тренування дискримінатор більше не потрібний, достатньо використовувати енкодер-декодер мережу стандартним чином.

Порівняльний аналіз робіт

Перш за все, слід зазначити, що усі три розглянуті задачі (виправлення помилок, спрощення тексту, перефразування) є прикладами умовної генерації тексту. А тому архітектурно це дуже схожі підходи – більшість моделей це енкодер-декодер трансформери, а тому велика частина процесу тренування і розробки виглядає дуже схоже.

Неочікувано більшість state-of-the-art моделей не використовують великі мовні моделі чи просто моделі з кількістю параметрів більшою за 1 мільярд. Натомість, вони досягають потужних результатів за рахунок підлаштування до задачі – десь ефективне використання даних специфічне для конкурентного завдання, десь побудова

контрольованих конфігурацій моделей для покращення цільової метрики у тестовому наборі даних, деє – використання додаткових знань (словників) або додаткових складових у тренуванні моделей.

Тобто важливу роль грає саме доменне знання, у комбінації з класичним на сьогодні підходом – тренування seq2seq трансформерів, потенційно попередньо натренованих для більш ефективного тренування. В той же час рішення про передову модель приймаються на базі автоматичних метрик, які не виглядають дуже надійними. Тому варто ставитися до таких результатів обережно, оскільки моделі могли просто добре підлаштуватися спеціально під тестові дані.

Список використаних джерел

1. *Ouyang L. et al.* Training language models to follow instructions with human feedback – Режим доступу до статті: <https://arxiv.org/abs/2203.02155>
2. *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>
3. *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>
4. *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>
5. *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>
6. *Dahlemeier D., Ng H.W.* Better Evaluation for Grammatical Error Correction – Режим доступу до статті: <https://www.aclweb.org/anthology/N12-1067.pdf>
7. *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>
8. *Zhao W. et al.* Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data – Режим доступу до статті: <https://www.aclweb.org/anthology/N19-1014.pdf>
9. *Stahlberg F. and Kumar S.* Synthetic Data Generation for Grammatical Error Correction with Tagged Corruption Models – Режим доступу до статті: <https://www.aclweb.org/anthology/2021.bea-1.4.pdf>
10. *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>
11. *Zhang X., Lapata M.* Sentence Simplification with Deep Reinforcement Learning – Режим доступу до статті: <https://www.aclweb.org/anthology/D17-1062.pdf>
12. *Xu W. et al.* Optimizing Statistical Machine Translation for Text Simplification – Режим доступу до статті: <https://www.aclweb.org/anthology/Q16-1029.pdf>
13. *Kincaid J.P. et al.* Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel
14. *Zhao S. et al.* Integrating Transformer and Paraphrase Rules for Sentence Simplification – Режим доступу до статті: <https://www.aclweb.org/anthology/D18-1355.pdf>
15. *Pavlick E. and Callison-Burch C.* Simple PPDB: A Paraphrase Database for Simplification – Режим доступу до статті: <https://www.aclweb.org/anthology/P16-2024.pdf>
16. *Omelianchuk K. et al.* Text Simplification by Tagging – Режим доступу до статті: <https://arxiv.org/pdf/2103.05070.pdf>
17. *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>
18. *Yang Z. et al.* XLNet: Generalized Autoregressive Pretraining for Language Understanding Rewrite – Режим доступу до статті: <https://arxiv.org/pdf/1906.08237.pdf>
19. *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>
20. Quora Duplicate Questions | Kaggle [Електронний ресурс] – Режим доступу:

- <https://www.kaggle.com/aymenmouelhi/quora-duplicate-questions>
21. Lin T. et al. Microsoft COCO: Common Objects in Context – Режим доступу до статті: <https://arxiv.org/pdf/1405.0312.pdf>
 22. 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>
 23. Satanjeev B., Alon L. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments – Режим доступу до статті: <https://aclanthology.org/W05-0909/>
 24. Guo Y. et al. Zero-Shot Paraphrase Generation with Multilingual Language Models – Режим доступу до статті: <https://arxiv.org/pdf/1911.03597.pdf>
 25. Parto B. N. et al. Learning Semantic Sentence Embeddings using Pair-wise Discriminator Models – Режим доступу до статті: <https://arxiv.org/pdf/1806.00807.pdf>
- Error Correction with Tagged Corruption Models. In *BEA 2021*
10. XU, W. et al. (2015) Problems in Current Text Simplification Research: New Data Can Help. In *TACL 2015*
 11. ZHANG, X. and LAPATA, M. (2017) Sentence Simplification with Deep Reinforcement Learning. In *EMNLP 2017*
 12. XU, W. et al. (2016) Optimizing Statistical Machine Translation for Text Simplification. In *TACL 2016*
 13. KINCAID, J. P. et al. (1975) *Derivation Of New Readability Formulas*. Institute for Simulation and Training, 56
 14. ZHAO, S. et al. (2018) Integrating Transformer and Paraphrase Rules for Sentence Simplification. In *EMNLP 2018*
 15. PAVLICK, E. and CALLISON-BURCH, C. (2016) Simple PPDB: A Paraphrase Database for Simplification. In *ACL 2016*
 16. OMELIANCHUK, K. et al. (2021) Text Simplification by Tagging. In *BEA 2021*
 17. OMELIANCHUK, K. et al. (2019) GECToR – Grammatical Error Correction: Tag, Not Rewrite. In *BEA 2019*
 18. YANG, Z. et al. (2019) XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NeurIPS 2019*
 19. WIETING, J. and GIMPEL, K. (2017) PARANMT-50M: Pushing the Limits of Paraphrastic Embeddings with Millions of Machine Translations. In *ACL 2017*
 20. KAGGLE. (2017) *Quora Duplicate Questions* [Online] – Available from: <https://www.kaggle.com/aymenmouelhi/quora-duplicate-questions> [Accessed: 19th June 2012].
 21. LIN, T. et al. (2014) Microsoft COCO: Common Objects in Context. In *ECCV 2014*
 22. PAPANENI, K. et al. (2002) Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL 2002*
 23. SATANJEEV B., ALON L. (2005) METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *ACL 2005*
 24. GUO, Y. et al. (2019) Paraphrase Generation with Multilingual Language Models. In *ACL 2019*
 25. PARTO, B. N. et al. (2018) Learning Semantic Sentence Embeddings using Pair-wise Discriminator Models. In *COLING 2019*

References

1. OUYANG L. et al. (2022) Training language models to follow instructions with human feedback In *NeurIPS 2022*
2. BRYANT, C. et al. (2019) The BEA-2019 Shared Task on Grammatical Error Correction. In *ACL 2019*
3. DAHLMEIER, D. and NH, T. H. (2012) Better Evaluation for Grammatical Error Correction. In *NAACL 2012*
4. TAJIRI, T. et al. (2012) Tense and Aspect Error Correction for ESL Learners Using Global Context. In *ACL 2012*
5. YANNAKOUDAKIS, H. et al. (2011) A New Dataset and Method for Automatically Grading ESOL Texts. In *ACL 2011*
6. DAHLMEIER, D. and NH, T. H. (2012) Better Evaluation for Grammatical Error Correction. In *NAACL 2012*
7. BRYANT, C. et al. (2017) Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *ACL 2017*
8. ZHAO, W. et al. (2019) Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data. In *NAACL 2019*
9. STAHLBERG F. and KUMAR S. (2021) Synthetic Data Generation for Grammatical

Надійшла до редакції 11.10.2023