ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ
INTELLIGENT SYSTEMS

УДК 004.8
https://doi.org/10.37661/1816-0301-2026-23-1-26-38

Поступила в редакцию | Received 11.02.2026
Подписана в печать | Accepted 03.03.2026
Опубликована | Published 31.03.2026

BelLitGPT – технологии языковых моделей для белорусского языка

Д. А. Ляхов, А. М. Бондоловский✉, С. В. Кругликов, К. К. Шульган
✉E-mail: a.bandalouski@newman.bas-net.by

Объединенный институт проблем информатики
Национальной академии наук Беларуси,
Сурганова, 6, Минск, 220012, Беларусь

Аннотация

Цели. Работа выполнена в области исследования специализированных генеративных нейронных сетей для белорусского языка. Поставлена цель сделать первый шаг для построения национальной генеративной языковой модели.

Методы. Описывается процесс разработки модели BelLitGPT (700 млн параметров), который основан на стратегии трансферного обучения русскоязычной модели ruGPT-3 и состоит из трех этапов: подготовки корпуса, адаптации токенизатора и обучения модели. Обучающий корпус составлен из золотого фонда классической белорусской прозы и подготовленных статей из Википедии. Подробно описываются методика адаптации токенизатора для расширения словарного запаса специфическими белорусскими лексемами, процесс обучения и тестирования модели.

Результаты. Результаты исследования подтверждают способность модели BelLitGPT генерировать связные, грамматически и стилистически корректные тексты. Особое внимание уделено созданию гибридного нейросимвольного подхода для генерации четверостиший с соблюдением ритма и рифмы.

Заключение. Эксперимент по масштабированию архитектуры показал сложности в обучении крупной модели (13 млрд параметров) в условиях дефицита данных.

Ключевые слова: большие языковые модели, трансферное обучение, нейросимвольный подход, генерация стихов, модель BelLitGPT, белорусский язык

Благодарности. Авторы выражают благодарность следующим сотрудникам ОИПИ НАН Беларуси: старшему научному сотруднику лаборатории распознавания и синтеза речи Н. В. Супрунчуку за проверку текстового корпуса, младшему научному сотруднику названной лаборатории А. Е. Дрогун за помощь в сборе и подготовке данных, заместителю генерального директора по научной работе С. Н. Касанину за успешные переговоры по получению в безвозмездное пользование вычислительного устройства AMD Ryzen AI Max+ PRO 395 на время проведения эксперимента и заведующему лабораторией высокопроизводительных систем О. П. Чижу за содействие в настройке устройства.

Для цитирования. BelLitGPT – технологии языковых моделей для белорусского языка / Д. А. Ляхов, А. М. Бондоловский, С. В. Кругликов, К. К. Шульган // Информатика. – 2026. – Т. 23, № 1. – С. 26–38. – https://doi.org/10.37661/1816-0301-2026-23-1-26-38.

Конфликт интересов. А. М. Бондоловский является членом редакционной коллегии журнала «Информатика» с 2026 г., но не имеет никакого отношения к решению опубликовать эту статью. С. В. Кругликов является членом редакционной коллегии журнала «Информатика» с 2016 г., но не имеет никакого отношения к решению опубликовать эту статью. Статья прошла принятую в журнале процедуру рецензирования. Об иных конфликтах интересов авторы не заявляли.

BelLitGPT – language model technologies for the Belarusian language

Dmitry A. Lyakhov, Andrei M. Bandalouski✉, Sergey V. Kruglikov, Konstantin K. Shulgan
✉E-mail: a.bandalouski@newman.bas-net.by

The United Institute of Informatics Problems of the National Academy of Sciences of Belarus, st. Surganova, 6, Minsk, 220012, Belarus

Abstract

Objectives. The research is conducted in the field of specialized generative neural networks for the Belarusian language. The authors aim to take the first step towards building a national generative language model.

Methods. The paper describes the development process of the BelLitGPT model (700 million parameters). It is based on a transfer learning strategy using the Russian-language model ruGPT-3 and consists of three stages: corpus preparation, tokenizer adaptation methodology and model training. The training corpus is compiled from the golden fund of classic Belarusian prose and prepared Wikipedia articles. The paper details the tokenizer adaptation method for expanding the vocabulary with specific Belarusian lexemes, as well as the model training and testing process.

Results. The research results confirm that BelLitGPT can generate coherent, grammatically and stylistically correct texts. Special attention is given to the creation of a hybrid neuro-symbolic approach for generating quatrains that adhere to rhythm and rhyme.

Conclusion. The experiment on scaling the architecture revealed difficulties in training a large model (13 billion parameters) under conditions of data scarcity.

Keywords: large language models (LLM), transfer learning, neuro-symbolic approach, poetry generation, model BelLitGPT, Belarusian language

Acknowledgments. The authors express their gratitude to the following coworkers of the UIIP NAS of Belarus: Mikita V. Suprunchuk, Senior Researcher at the Laboratory of Speech Synthesis and Recognition, for verifying the text corpus; Anastasia E. Drogun, Junior Researcher at the Laboratory of Speech Synthesis and Recognition, for assistance with data collection and preparation; Sergey N. Kasanin, Deputy General Director for Research, for successful negotiations regarding the free use of an AMD Ryzen AI Max+ PRO 395 computing device for the duration of the experiment; and Oleg P. Chizh, Head of the High-Performance Systems Laboratory, for assistance with device setup.

For citation. Lyakhov D. A., Bandalouski A. M., Kruglikov S. V., Shulgan K. K. *BelLitGPT – language model technologies for the Belarusian language*. Informatika [*Informatics*], 2026, vol. 23, no. 1, pp. 26–38 (In Russ.). https://doi.org/10.37661/1816-0301-2026-23-1-26-38.

Conflict of interests. A. M. Bandalouski has been a member of the editorial board of the journal "Informatics" since 2026 but had no role in the decision to publish this article. S. V. Kruglikov has been a member of the editorial board of the journal "Informatics" since 2016 but had no role in the decision to publish this article. The article has undergone the journal's established peer-review process. The authors declare of no other conflicts of interest.

Введение

Последнее десятилетие ознаменовалось стремительным развитием области обработки естественного языка (англ. natural language processing, NLP), которое стало возможным благодаря появлению и широкому распространению больших языковых моделей (БоЯМ, англ. large language models, LLM) [1–3]. Эти модели обладают хорошими способностями в создании текстов, выполнении переводов и решении интеллектуальных задач. Тем не менее, несмотря на значительные международные достижения в данной

области, большинство современных решений остаются ориентированными преимущественно на английский язык и ограниченное число наиболее распространенных мировых языков. Для языков с небольшим объемом доступных цифровых материалов, включая белорусский, существующие технологии часто показывают недостаточно высокое качество генерации, склонны порождать артефакты (так называемые галлюцинации) и недостаточно хорошо понимают специфику культурного контекста. В связи с этим создание национальной большой языковой модели не только является технической задачей, но и становится важнейшей предпосылкой для поддержания культурной самобытности и укрепления цифрового суверенитета в изменяющемся мире. Научная область исследований специализированных генеративных нейронных сетей для белорусского языка новая и малоизученная. Можно сказать, что на сегодняшний день практически отсутствуют открыто обученные модели, которые показывают высокое владение белорусским языком.

Данная работа призвана преодолеть указанный недостаток и положить начало систематическим исследованиям в области создания национальной БоЯМ. Исследование рассматривается как первый шаг на пути к построению развитой цифровой экосистемы сервисов генеративного белорусского искусственного интеллекта (ИИ). Под «экосистемой белорусского ИИ» понимается система решений, которые выстроены на одной платформе вокруг мощной базовой языковой модели, обладающей знанием белорусского языка и национального контекста. К окружающим «ядро» (языковую модель) решениям могут относиться такие сервисы и надстройки, как поиск в интернете, управление подсказками при запросах, обеспечение безопасности содержания, диалоговые интерфейсы для конечных пользователей (чат-боты), API для интеграции в сторонние приложения, специализированные ИИ-ассистенты (для образования, анализа документов, юридической или медицинской сферы) и др.

В качестве отправной точки для экспериментов была выбрана стратегия трансферного обучения [4]. В статье описываются шаги ее осуществления, в частности процесс адаптации и дообучения модели на специально собранном корпусе белорусских текстов. Авторы предположили, что нативные русскоязычные модели наиболее легко и успешно переобучаются белорусскому языку, и выбрали предварительно обученную малую языковую модель ruGPT-3[1] с 700 млн параметров (далее – ruGPT-3). Благодаря заложенным знаниям в языковой модели с родственным языком выбранный подход позволяет эффективно начать разработку суверенной БоЯМ даже при ограниченных вычислительных ресурсах и данных.

1. Подготовка данных

Цель подготовительных работ – создание компактного набора данных, который подходит для переобучения современных языковых моделей, приведен к единому орфографическому стандарту и содержит в себе признанные социокультурные ценности белорусского народа. Далее опишем процесс создания текстового корпуса для переобучения выбранной модели.

[1]SberDevices, Sber AI, and SberCloud. rugpt-3: Open source russian gpt-3 models. GitHub repository, 2020.

Корпус включил в себя произведения классической белорусской литературы и статьи из белорусской Википедии. Основной набор данных корпуса сформирован из золотого фонда классической белорусской прозы. В него включены труды 20 века следующих авторов:

Уладзімір Караткевіч	Міхась Лынькоў
Якуб Колас	Алесь Адамовіч
Іван Шамякін	Кузьма Чорны
Янка Брыль	Эліза Ажэшка
Кандрат Крапіва	Зьмітрок Бядуля
Васіль Быкаў	Цішка Гартны
Пятрусь Броўка	Максім Гарэцкі
Іван Навуменка	Алесь Савіцкі
Ніл Гілевіч	Цётка
Андрэй Макаёнак	Міхась Чарот
Іван Чыгрынаў	Ядвігін Ш.
Іван Мележ	Віктар Супрунчук

Весь массив текстов был выверен в соответствии с официальными правилами белорусской орфографии 1957 г. (утверждены постановлением Совета министров БССР)[2]. Для этого вначале корпус прошел проверку правописания с помощью мультиязычной БоЯМ, а затем был дополнительно утвержден кандидатом филологических наук, старшим научным сотрудником лаборатории распознавания и синтеза речи ОИПИ НАН Беларуси Н. В. Супрунчуком. Общий объем текстов авторов составил 5 250 182 слова (около 59 МБ).

В то же время литература классических авторов не всегда отражает реалии современного мира и не содержит новые понятийные категории. В связи с этим для расширения лексического разнообразия корпуса и повышения качества и уровня связности текстов [5] будущей малой белорусской языковой модели BelLitGPT в работе были использованы материалы Википедии. Каждая статья прошла процедуру очистки и фильтрации. В результате был получен текстовый файл с отобранными статьями на белорусском языке. Вместе с прозой классиков он сформировал объединенный корпус белорусских текстов объемом 6 333 013 слов (около 75 МБ).

Следует отметить, что сохранение социокультурных ценностей в ответах белорусской генеративной модели в значительной мере достигается через отбор информационного содержания корпуса. Особенность генеративных трансформеров заключается в том, что нормы языка передаются модели через корпус текстов при обучении, поэтому необходимо подготовить тексты, соответствующие языковому стандарту. Это является важной предпосылкой для устойчивого качества генерируемых текстов.

[2]Отобранные произведения написаны до вступления в силу действующего Закона «О правилах белорусской орфографии и пунктуации» от 23 июля 2008 г. Различия между правилами 1957 и 2008 гг. незначительные.

2. Создание белорусского токенизатора

Предложенный подход к созданию белорусского токенизатора основывается на процедуре расширения словарного запаса (токенизатора) модели ruGPT-3. Русскоязычные токенизаторы могут разбивать слово на несколько токенов, что позволяет адаптировать их к белорусскому языку, в то время как англоязычные одному слову сопоставляют один токен [6].

Белорусский язык отличается от русского по фонетике, лексике, морфологии, орфографии, синтаксису. Между тем базовая механика токенизации – это только разделение слов на токены, поэтому при разработке белорусского токенизатора не требуется учитывать все отличия между языками. Поскольку оба языка используют кириллицу, базовый русскоязычный токенизатор изначально способен обрабатывать белорусскую лексику, разбивая незнакомые слова на более короткие известные участки (субтокены). Однако отсутствие в словаре токенизатора незнакомых букв «ў», «і» и знака «'» (апостроф) приводит к ошибкам в процессе работы. Находя их, он создает токены <unk> (unknown), наличие которых делает дальнейшее обучение невозможным. Добавление указанных символов является минимально необходимым техническим шагом (первым этапом), после которого токенизатор способен обрабатывать белорусские тексты. Для этого авторы создают временный BPE(Byte Pair Encoding)-токенизатор и обучают его брать частые слова целиком и трактовать их как один токен, а редкие или незнакомые разбивать на части [7]. Например, слово ’слоўнік’ базовый токенизатор может разбить на неэффективные фрагменты [’сло’, ’ў’, ’н’, ’і’, ’к’]. В то же время адаптированный токенизатор может распознавать семантическое ядро слова (корень) и его грамматические модификаторы (приставки и суффиксы) как отдельные смысловые единицы. В результате нейросеть обучается не просто запоминать слова целиком, а выводить значение новых или редких терминов, опираясь на знакомые морфемы, что критически важно для белорусского языка с его богатой системой словообразования.

Стоит отметить, что между белорусским и русским языками существуют и другие алфавитные различия, однако они не влияют на работоспособность токенизатора. Так, присутствие в словаре русской буквы «щ» не снижает точности токенизации. Избыточные для белорусского языка символы игнорируются при обработке и не влияют на результат, а при обработке белорусских «дз» и «дж» модель может разделять названные диграфы на разные субтокены, однако это не сказывается на связности и правильности генерируемого текста.

На следующем шаге алгоритм вычисляет разность множеств между словарями исходной модели и новой. Так, он выделяет токены, которые присутствуют в белорусском словаре, но отсутствуют в русском. Далее из этого списка отбирается фиксированное количество наиболее важных токенов, в данной конфигурации 3000. Ограничение в 3000 токенов выбрано как компромисс между полнотой охвата лексики и вычислительной сложностью модели. В список попадают наиболее частотные белорусские лексемы (например, ’гэта’, ’ёсць’, ’праз’) и специфические морфемы, содержащие буквы «і» и «ў». Это дает возможность BelLitGPT воспринимать характерные белорусские слова, например с «ў» и «і», целиком, а не разбивать их на случайные буквы или слоги. Таким образом, представленная методика позволила создать белорусский токенизатор. Его использование повышает качество генерации текста.

В свою очередь, расширение словаря привело к необходимости обновления параметров самой нейросети. Для этого веса из исходной модели ruGPT-3 загрузили в нейросеть и произвели изменение размерности входного и выходного слоев в соответствии с размером нового словаря. Так была получена базовая белорусская модель rugpt-belarusian-base, которая готова для обучения на белорусском корпусе текстов.

3. Выстраивание процесса обучения

Процесс обучения построен на базе библиотеки Hugging Face Transformers и включает в себя четыре ключевые стадии: инициализацию, подготовку данных, настройку гиперпараметров и цикл оптимизации.

В начале работы система загружает базовую белорусскую модель rugpt-belarusian-base и соответствующий ей токенизатор. Для возможности обучения крупной модели на ограниченных аппаратных ресурсах NVidia RTX 4090 активируется механизм контрольного сохранения градиентов. Это позволяет существенно экономить видеопамять (VRAM) за счет пересчета части градиентов «на лету» вместо их хранения, что в свою очередь дает возможность использовать более крупные пакеты данных.

Далее происходят предварительная обработка и сегментация данных. Действия направлены на оптимизацию эффективного использования контекстного пространства модели и являются наиболее ресурсоемким этапом процесса обучения. Объединенный корпус белорусских текстов подвергается процедуре преобразования в последовательный ряд числовых токенов. Обучение осуществляется не на изолированных предложениях, а на объединенном длинном потоке текста, который разделяется на фрагменты постоянной протяженности (фиксированная длина блока составляет 1024 токена). Это обеспечивает подачу на вход модели максимального объема контекста, который препятствует заполнению пустых позиций, исключает потерю данных на стыках предложений и увеличивает производительность обучения.

Для более эффективной перенастройки модели под новый язык были установлены следующие гиперпараметры обучения:

Скорость обучения. Установлено значение $3 \cdot 10^{-4}$, что несколько выше стандартного для дообучения. Это сделано намеренно для более агрессивной адаптации к новому языковому распределению.

Эпохи. Процесс подразумевает пять полных проходов по всему набору данных.

Смешанная точность. Используются 16-битные числа (FP16) с плавающей запятой для ускорения вычислений и снижения потребления памяти.

Накопление градиента. Позволяет эмулировать большой размер пакетов путем накопления результатов нескольких шагов перед обновлением весов модели.

Затем инициируется итерационный алгоритм оптимизации целевой функции потерь. Автоматизированная система осуществляет управление загрузкой тренировочных данных и регулярное обновление весовых коэффициентов нейронной сети.

По окончании указанного цикла оптимизационного процесса финальная версия обученной модели вместе с откорректированным лексиконом токенизатора сохраняется в каталоге BelLitGPT.

Итоговым результатом реализации указанного выше процесса переобучения становится полнофункциональная генеративная модель. Она обладает способностью создавать осмысленные и согласованные тексты на белорусском языке, которые отражают особенности исходного корпуса золотого фонда классической белорусской прозы.

4. Тестирование модели

Для валидации обученной модели BelLitGPT была проведена серия качественных экспериментов. Основной целью тестирования являлась проверка способности модели генерировать текст, соответствующий не только грамматическим и морфологическим нормам, но и семантике белорусского литературного языка.

Тестирование проводилось методом zero-shot-генерации. Его суть состоит в том, что в модель подается начальная фраза (промпт). После генерации ответа оценивается лингвистическая корректность продолжения. Ниже представлены результаты проверки базовых лингвистических компетенций.

- **Тест 1: грамматическое согласование (падежные формы)**

  Вход. Сонца павольна схавалася за [...]

  Выход: лесам.

  Анализ. Модель корректно определила управление предлога *за* (творительный падеж) и сгенерировала существительное в правильной словоформе, что подтверждает усвоение структуры падежных окончаний.

- **Тест 2: морфологическое согласование (род)**

  Вход. Пасярод поля цякла глыбокая [...]

  Выход: рачулка.

  Анализ. Прилагательное *глыбокая* указывает на женский род. Модель успешно согласовала с ним сгенерированное существительное *рачулка*, игнорируя частотные варианты мужского или среднего рода, что говорит о корректном моделировании зависимостей внутри именной группы.

- **Тест 3: логическая и контекстуальная связность**

  Вход. Неба раптам пацямнела, і пачаўся моцны [...]

  Выход: вецер.

  Анализ. В связи с сопряженностью таких природных явлений, как дождь и ветер во время штормовой погоды, продолжением высказывания может быть и «дождь». Однако, если углубиться в физику природных явлений, то резкое потемнение неба (формирование кучево-дождевой облачности) перед грозой чаще всего сопровождается именно шквалистым усилением ветра, которое нередко предшествует самому дождю. Поэтому в рассматриваемом случае модель продемонстрировала понимание причинно-следственных связей.

В контексте оценки генеративных способностей БоЯМ важно учитывать, что они являются вероятностными, предсказывающими следующий токен на основе распределения вероятностей, сформированного в процессе обучения. Понимание причинно-следственных связей в их работе проявляется в том, что алгоритм снижает вероятность логически несовместимых токенов и выбирает продолжение из семантически согласованного кластера (ветер, дождь, ливень, шторм, ураган). Тот факт, что модель выбирает корректное метеорологическое следствие из вероятностного распределения, свидетельствует

о наличии «здравого смысла» (common sense reasoning) в латентном пространстве модели и, в частности, о том, что она успешно усвоила статистические признаки и контекстуальные зависимости, характерные для описания непогоды. При этом речь идет не о понимании в человеческом смысле, а о согласованном вероятностном прогнозировании на основе выявленных закономерностей в данных.

Приведем дополнительные примеры, которые показывают способность белорусской модели генерировать текст с логической и контекстуальной связностью.

Пример 1.
Вход. Каб адчыніць замкнёныя дзверы, ён дастаў з кішэні […]
Выход: ключ і пачаў круціць.

Пример 2.
Вход. На вуліцы быў моцны мароз, таму вада ў лужынах хутка […]
Выход: замярзала.

Пример 3.
Вход. Калі яна пачула гэтую радасную навіну, на яе твары з'явілася […]
Выход: ўсмешка.

Пример 4.
Вход. Каб купіць свежага хлеба і малака, яна ўзяла торбу і пайшла ў […]
Выход: мястэчка.

- **Тест 4: лексическая сочетаемость**
  Вход. Аўтар у сваім творы хацеў паказаць жыццё простым і [...]
  Выход: ясным.

Анализ. Модель восстановила устойчивую литературную коллокацию *простым і ясным*. Это свидетельствует о глубоком влиянии корпуса художественной литературы, так как подобные обороты характерны для публицистики и литературной критики.

Если первая серия тестов подтвердила компетенцию модели на уровне предложения, то следующий этап валидации был направлен на оценку генеративных способностей на макроуровне (уровне связного текста). Проведенные испытания показали способность BelLitGPT развивать контекст через добавление новых персонажей, описание природы и включение монологов в созданном тексте из нескольких предложений. Приведем пример генерации текста с входящим запросом «Стары замак стаяў на гары...» и оценим его.

Промпт. Стары замак стаяў на гары […]

Результат:

...і з замкавай вежы можна было бачыць увесь горад. А калі ўздымалася сонца над горадам, то яно стаяла высока, і здавалася, што гэта сам Стары Замак відзён з вышыні. У такія хвіліны сэрцу робіцца лёгка ад таго, што ўсё навокал свеціцца залатым бляскам. І ў гэтыя хвіліны яснай летняй раніцы Васіль думаў пра свой родны кут: які ён бедны! Хто б мог падумаць...

Проведем комплексный стилистико-семантический анализ сгенерированного фрагмента:

Семантическое расширение. Модель логично развила пространственную сцену: от замка на горе до панорамы города, видимой с башни.

Эмоциональная окраска. Текст содержит сложные синтаксические конструкции, описывающие внутреннее состояние (*сэрцу робіцца лёгка*), что свойственно художественной прозе.

Стилистическая мимикрия. Появление лирического героя (Васіля) и резкий переход к социальной рефлексии (*які ён бедны!*) ярко демонстрируют влияние обучающего корпуса классиков (Я. Коласа, К. Чорного), для которых характерно переплетение пейзажной лирики с социальными мотивами.

5. Нейросимвольный подход для генерации четверостиший

Для дополнительной иллюстрации способностей BelLitGPT генерировать текст, придерживаясь стилистических и семантических правил, был рассмотрен особый случай – создание стихов. Для этого использовалась архитектура гибридного конвейера, основанная на нейросимвольном подходе и объединяющая нейросетевые модели генерации текста, жесткие алгоритмические ограничения (задающие правила ритма и рифмы) и верификацию с помощью БоЯМ, которая играет роль судьи [8].

Для того чтобы BelLitGPT стала генерировать четверостишия, ее дообучили на поэме Якуба Коласа «Новая Зямля». Для этого из всей поэмы было составлено множество пар «запрос – ответ», где запросом является первая строка четверостишия, а ответом – три остальные. Далее модель BelLitGPT дообучили на всем множестве пар. Говоря про процедуру дообучения моделей, исследователи отмечают, что она базируется на единой методологии непрерывного дообучения (continuous fine-tuning) и является универсальной для всех языков [4]. Отличиям же подвержены архитектурные особенности построения моделей с использованием нейросимвольного подхода [8, 10, 11].

Процесс дообучения строится на подаче структурированных пар данных, обрамленных специальными токенами: <|startoftext|> для обозначения начала, <|sep|> для разделения запроса и ответа и <|endoftext|> для фиксации конца последовательности. На каждой итерации модель генерирует предсказание следующего токена, которое сопоставляется с эталонным текстом для вычисления функции потерь. Через алгоритм обратного распространения ошибки выявленное отклонение корректирует внутренние веса нейросети, что постепенно минимизирует влияние общего корпуса и адаптирует модель к особенности целевого текста (в данном случае к стихам в размере четырехстопного ямба). С формальной точки зрения вышеуказанный алгоритм минимизирует функцию потерь:

$$L = -\sum_{i=1}^{n} \log P(y_i \,|\, x, y_{<i}),$$

где x – первая строка, y_i – токены трех последующих строк.

Первая попытка генерации продолжения по одной заданной строчке оказалась неудачной. Соблюдение ритма и рифмы было скорее исключением, чем правилом. Например, на 1000 попыток со строкой «Гляджу ў акно на ціхі сад» нашлась только одна со схемой рифмовки AABB[3].

Пример 1.

Гляджу ў акно на ціхі сад,
Нібы пытае воўк цыкад…

[3]Схема построения строфы, при которой рифмуются две соседние строки (первая со второй, третья с четвертой). Пример: Мой родны кут, як ты мне мілы! (А) Забыць цябе не маю сілы! (А) Не раз, утомлены дарогай, (В) Жыццём вясны мае убогай… (В)

Поэтому для успешного создания четверостиший было предложено использовать генератор в двух разных режимах: в прямой генерации и обратной.

Прямая генерация используется для создания трех последующих строк по одной данной первой строке. Модель предсказывает продолжение текста на основе контекста с изменяющейся для повышения вариативности температурой воспроизведения от 0,9 до 1,3.

Обратная генерация применяется для строк, которые требуют соблюдения рифмы. Алгоритм сначала выбирает целевое рифмующееся слово из словаря, а затем генерирует предшествующий ему текст, обеспечивая когерентность всего четверостишия.

В случае обратной генерации необходимо следить, чтобы в созданном ответе сохранялся смысл. Для этого используется более совершенная языковая модель Gemma 2 (с 9 млрд параметров) [9]. Она оценивает создаваемые строки с точки зрения смысла $Score_{sense}$ и грамматики $Score_{gram}$ и выбирает ту, у которой лучшие оценки. Такой подход соответствует стандартной для машинного обучения схеме «генератор – дискриминатор» [10].

Между тем для обеспечения ритма и рифмы также требуется специальный детерминированный модуль. Он был построен с применением фонетического анализа, который совершается за счет проставленных ударений в грамматической базе данных ресурса bnkorpus.info[4]. Модуль допускает строки длиной восемь или девять слогов с чередованием ударных и безударных позиций (хорей/ямб) [11]. Рифма определяется по совпадению фонетических концовок слов (ударная гласная и последующие звуки) с учетом вариативности гласных, например *ы/і*.

5.1. Алгоритм генерации четверостишия

Генерация строфы по схеме AABB реализуется следующим образом:

1. На вход подается инициирующая строка L_1.
2. Для генерации парной строки L_2 производится поиск множества рифм $R = \{r_1, r_2, \ldots, r_n\}$ к последнему слову L_1.
3. Запускается итеративный процесс (до 100 попыток), в котором генератор создает кандидатов, заканчивающихся на $r_i \in R$.
4. Кандидаты фильтруются по метрике и передаются модулю валидации. Лучший кандидат по сумме баллов ($Score_{gram} + Score_{sense}$) принимается как L_2.
5. Аналогичный процесс повторяется для пары строк L_3 и L_4, где L_3 генерируется в режиме прямой генерации для развития сюжета.

Пример 2.
Гарыць агонь у печы зноў
І бліскае там язычкоў
Над квольм лісцем, над асінай
Ды над кудзеляй павучынай

Пример 3.
Гучыць жалейка над ракой
І плача бедная святой
Над гэтай гразёй, над балотам
І над табою абармотам

[4]Нацыянальны корпус беларускай мовы. – URL: http://bnkorpus.info (дата обращения: 26.02.2026). Грамматическая база создана, поддерживается и обновляется сотрудниками сектора компьютерной лингвистики ГНУ «Центр исследований белорусской культуры, языка и литературы НАН Беларуси».

Пример 4.
Вятрыска гне стары дубок
І буйны ветрык-вечарок
Над гэтым лесам парыўся
І мёдам росным марыўся

Пример 5.
Снягі ўкрылі ціхі сад
І ночкі снегам неўпапад
Па лесе разышліся, хвалі
І белым пухам дабягалі

Рассмотрим отличия предложенного авторами нейросимвольного подхода от других существующих в литературе. Так, в модели Neural Poetry [11] входными токенами выступают исключительно слоги. В нейросимвольном подходе применяется процедура расширения словарного запаса BPE-токенизатора, что позволяет нейросети распознавать семантическое ядро слова (корень) и его грамматические модификаторы (приставки и суффиксы) как отдельные смысловые единицы. Это критически важно для корректной работы с богатой системой словообразования.

Модель Deep-speare [10] решает задачу полностью нейросетевым способом, обучаясь ритму и рифме без внешних словарей и эвристик. Система Hafez [8] использует конечный автомат, который жестко ограничивает пространство на этапе лучевого поиска. В архитектуре гибридного конвейера используется специальный детерминированный фонетический модуль на основе грамматической базы белорусского языка (bnkorpus.info), который отбирает кандидатов с длиной восемь-девять слогов и чередованием ударных (безударных) позиций.

Если Neural Poetry оценивает сгенерированные пакеты с помощью математических штрафов за нарушение формы, а Deep-speare и Hafez полагаются на вероятности своих внутренних рекуррентных моделей, то авторы применяют внешнюю языковую модель в роли независимого судьи. Она оценивает кандидатов (в частности, при обратной генерации) с точки зрения сохранения смысла и грамматики, реализуя соревновательную схему «генератор – дискриминатор».

Указанные отличия делают предложенную архитектуру оптимальной для работы с малыми языковыми моделями на вычислительных устройствах потребительского класса. Разработанный конвейер доказывает, что в условиях указанных ограничений можно успешно генерировать стихи, демонстрируя высокую связность текста и точное сохранение сложной поэтической структуры.

Заключение

В работе представлена методика создания национальной большой языковой модели BelLitGPT, основанная на стратегии трансферного обучения и адаптации токенизатора. Предложенный подход показал, что использование архитектуры модели (ruGPT-3) с родственным языком позволяет эффективно преодолевать нехватку подготовленных текстовых данных на белорусском языке.

Ключевым результатом исследования стала успешно натренированная модель BelLitGPT с 700 млн параметров. Эксперименты подтвердили, что даже при сравнительно небольшом объеме обучающего корпуса (6 333 013 слов или порядка 75 МБ), состоящего из произведений классической белорусской прозы и статей Википедии, модель способна усваивать грамматический строй, морфологические нормы и стилистические особенности белорусского языка. Качественная валидация показала способность

нейросети генерировать связные тексты, поддерживать контекст и демонстрировать элементы «здравого смысла». Кроме того, на базе полученной модели была успешно реализована система генерации четверостиший, которая использует нейросимвольный подход с верификацией ритма и рифмы.

Между тем в ходе исследования были выявлены и существенные ограничения, связанные с масштабированием архитектуры. Параллельный эксперимент по дообучению более крупной версии модели ruGPT-3 с 13 млрд параметров на вычислительном устройстве на базе процессора AMD Ryzen AI Max+ PRO 395 с 128 ГБ оперативной памяти LPDDR5x-8000 показал, что имеющегося объема текстовых данных недостаточно для качественной перенастройки столь массивной нейронной сети. Из-за дисбаланса между количеством обучаемых параметров и размером корпуса модель на 13В не смогла полностью переключиться на целевой язык. В ее генерациях сохранялось сильное влияние исходной модели, что выражалось в частом появлении русских слов и смешении грамматических правил.

Таким образом, для создания компактных моделей (до 1 млрд параметров) критически важными являются качество и чистота данных, в то время как для обучения крупных моделей (больше 10 млрд параметров) первоочередной задачей становится кардинальное увеличение объема корпуса. Собрать 1000 МБ подготовленных аутентичных белорусских текстов становится отдельной задачей, выполнение которой позволит приступить к обучению крупных моделей.

Вклад авторов. *А. М. Бондоловский* и *Д. А. Ляхов* внесли вклад в планирование исследования, анализ и интерпретацию данных и подготовку статьи. *С. В. Кругликов* и *К. К. Шульган* внесли вклад в подготовку статьи, осуществили редакцию статьи с точки зрения существенного интеллектуального содержания и утвердили окончательную версию статьи для публикации.

References

1. Brown T., Mann B., Ryder N., Subbiah M., Kaplan J. D., …, Amodei D. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 1877–1901.

2. Radford A., Wu J., Child R., Luan D., Amodei D., Sutskever I. Language models are unsupervised multitask learners. *OpenAI*, 2019. Available at: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf (accessed 03.11.2025).

3. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., …, Polosukhin I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017, vol. 30, pp. 5998–6008.

4. Artetxe M., Ruder S., Yogatama D. On the cross-lingual transferability of monolingual representations. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020*, pp. 4623–4637.

5. Jakubíček M., Kilgarriff A., Kovář V., Rychlỳ P., Suchomel V. The tenten corpus family. *Proceedings of the 7th International Corpus Linguistics Conference (CL2013), Lancaster University, United Kingdom, 22–26 July 2013*, pp. 125–127.

6. Sennrich R., Haddow B., Birch A. Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016*, vol. 1, pp. 1715–1725.

7. Imamura K., Sumita E. Vocabulary adaptation for domain adaptation in neural machine translation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October – 4 November 2018*, pp. 4623–4637.

8. Ghazvininejad M., Shi X., Choi Y., Knight K. Hafez: an interactive poetry generation system. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, 30 July – 4 August 2017*, pp. 43–48.

9. Mesnard T., Hardin C., Dadashi R., Bhupatiraju S., Pathak S., …, Eck D. *Gemma: Open models based on Gemini research and technology*, 2024. Available at: https://arxiv.org/pdf/2403.08295 (accessed 03.11.2025).

10. Lau J. H., Cohn T., Baldwin T., Brooke J., Hammond A. Deep-speare: A joint neural model of poetic language, meter and rhyme. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018*, vol. 1, pp. 1948–1958.

11. Zugarini A., Melacci S., Maggini M. Neural poetry: Learning to generate poems using syllables. *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series: 28th International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019*, pp. 313–325.

Информация об авторах

Ляхов Дмитрий Александрович, кандидат физико-математических наук, старший научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: dlyakhov@newman.bas-net.by

Андрей Михайлович Бондоловский, кандидат экономических наук, заведующий лабораторией распознавания и синтеза речи, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: a.bandalouski@newman.bas-net.by

Сергей Владимирович Кругликов, доктор военных наук, кандидат технических наук, доцент, главный научный сотрудник, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: kruglikov_s@newman.bas-net.by

Константин Константинович Шульган, заместитель генерального директора по цифровому развитию, Объединенный институт проблем информатики Национальной академии наук Беларуси.
E-mail: skk@newman.bas-net.by

Information about the authors

Dmitry A. Lyakhov, Cand. Sci. (Phys.-Math.), Senior Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: dlyakhov@newman.bas-net.by

Andrei M. Bandalouski, Cand. Sci. (Econ.), Head of Laboratory of Speech Synthesis and Recognition, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: a.bandalouski@newman.bas-net.by

Sergey V. Kruglikov, Dr. Sci. (Milit.), Cand. Sci. (Eng.), Assoc. Prof., Principal Researcher, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: kruglikov_s@newman.bas-net.by

Konstantin K. Shulgan, Deputy General Director for Digital Development, The United Institute of Informatics Problems of the National Academy of Sciences of Belarus.
E-mail: skk@newman.bas-net.by