

11) решение задачи должно осуществляться на базе глубокого и всестороннего предварительного анализа задачи, анализа хода решения, в том числе ретроспективного, поиска наиболее рационального решения;

13) привнесение элементов творчества в подходах к решению задач;

14) восприятие самим учителем простого и понятного для него задания как нового и удивительного, т.е. попытка восприятия проблемы глазами ученика.

### **Библиографический список**

1. Ивашова, О. А. ИКТ в становлении вычислительной культуры / О. А. Ивашова // Конгресс конференций «Информационные технологии в образовании». – URL: <http://ito.edu.ru/2006/Moscow/I/1/I-1-6499.html> (дата обращения: 30.10.2025)
2. Казакова, Т. Н. Методика формирования вычислительной культуры школьников пятых классов : лис. ... канд. пед. наук : 13.00.02. – Санкт-Петербург, 2004. – 172 с.
3. Калягин, Ю. М. Методика преподавания математики в средней школе. Общая методика: Учебное пособие для студентов физ.-мат. факультетов пед. институтов / Ю. М. Калягин, В. А. Оганесян, В. Я. Саннинский, Г. Л Луканкин. – Москва : Просвещение, 1975. – 462 с.
4. Козел, О. Н. Вычислительная компетентность будущих учителей информатики: структура, составляющие, формирование // Мир науки, культуры, образования. – 2009. – № 2 (14). – С. 204–207.
5. Общая психология : Учебник / Под общ. ред. проф. А. В. Карпова. – Москва : Гардарики, 2004. – 232 с.
6. Самарский, А. А. Математическое моделирование и вычислительный эксперимент / А. А. Самарский // Вестник АН СССР. – 1979. – № 5. – С. 38–49.
7. Файзиева, Д. Х. Анализ проблем изучения и преподавания программирования на уроках информатики // Universum : технические науки : электрон. научн. журн. – 2021. 12(93). – URL: <https://7universum.com/ru/tech/archive/item/12783> (дата обращения: 28.09.2025).
8. Чигиринская, Н. В. Вычислительное мышление будущего инженера: понятийный анализ и опыт формирования в техническом вузе / Н. В. Чигиринская, О. Е. Григорьева, А. М. Бочкин, М. И. Андреева // Современные научно-исследовательские технологии. – 2023. – № 2. – С. 205–211.

**Буландо Р.И., старший преподаватель кафедры японского языка института иностранных языков**

Московский государственный педагогический университет  
г. Москва, Россия

### **ДИДАКТИЧЕСКИЙ ПОТЕНЦИАЛ МАЛЫХ ЯЗЫКОВЫХ МОДЕЛЕЙ: ЛОКАЛЬНЫЙ АГЕНТ МАШИННОГО ПЕРЕВОДА В УЧЕБНОЙ САТ-СРЕДЕ**

**Аннотация.** Предлагается оффайн-first внедрение локального агента машинного перевода на базе малых языковых моделей (*Qwen 2.5, 1.5B/3B/7B*) в учебную САТ-среду *OmegаТ*. Описывается *portable-стек* для аудиторной работы без сети и облачных API, его встраивание в цикл «перевод – пост-редактирование – память переводов», а также план оценивания качества, производительности и образовательных результатов. Показано, что такой стенд превращает *LLM* из «чёрного ящика» в объект учебного действия: студент управляет архитектурой, данными и метриками, фиксируя воспроизводимость эксперимента и пределы автоматизации.

**Ключевые слова:** CAT-среда; локальный агент машинного перевода; малые языковые модели (SLM); офлайн-инференс; Qwen; OmegaT; нормализация китайского письма (OpenCC); параллельные корпусы (OPUS, UN); дообучение (LoRA/QLoRA); оценивание перевода (sacreBLEU)

## DIDACTIC POTENTIAL OF SMALL LANGUAGE MODELS: A LOCAL MT AGENT IN AN EDUCATIONAL CAT ENVIRONMENT

**Abstract.** *We present an offline-first deployment of a local machine-translation agent based on small language models (Qwen 2.5, 1.5B/3B/7B) integrated into the OmegaT CAT system for classroom use. A portable software stack enables work without internet or cloud APIs and fits the translate–post-edit–translation-memory cycle. We outline procedures for evaluating quality, speed, and learning outcomes. The setup turns the LLM from a “black box” into a teachable tool: students control architecture, data, and metrics, ensuring reproducibility and clarifying the limits of automation.*

**Keywords:** CAT environment; local machine translation agent; small language models (SLM); offline inference; Qwen; OmegaT; Chinese script normalization (OpenCC); parallel corpora (OPUS, UN); parameter-efficient finetuning (LoRA/QLoRA); translation evaluation (sacreBLEU)

В последние годы цифровая инфраструктура обучения переводу всё больше опирается на облачные сервисы [1, 2]. В аудитории это порождает эффект «чёрного ящика»: студент видит автоматически сгенерированный черновик перевода, но сам процесс его получения – выбор модели, параметры генерации, типичные ошибки – остаётся невидимым. Такая непрозрачность снижает наблюдаемость и управляемость инструмента со стороны обучающегося; между тем именно прозрачность и контроль над параметрами – необходимые условия для формирования метакогнитивных и пост-редакторских навыков (распознавание систематических ошибок, соотнесение настроек с качеством результата) [1, 3]. В CAT-сценарии это достигается, когда машинный перевод подключён как внешний сервис, а сама среда остаётся интерфейсом к памяти перевода и глоссариям, а не «скрытым переводчиком» [4].

Наш ответ на «чёрный ящик» – офлайн-first сборка локального агента машинного перевода (далее – локальный МП-агент) на базе малой языковой модели (SLM). Студент сам разворачивает, настраивает и подключает его к своей CAT-среде. Такой ход делает процесс наблюдаемым и управляемым: фиксируются версия модели и квантизация, явно задаются параметры инференса, системно замеряются метрики качества и производительности, а изменения сопоставляются с принятыми переводческими решениями [5]. При этом сохраняется привычный цикл работы переводчика: сегментация, поиск совпадений в памяти переводов (далее – ПП), обращение к терминологии, получение автоматического варианта, постредактирование и запись результата в ПП. Практическая часть проекта реализовывалась в рамках учебной дисциплины «Информационные технологии в лингвистике» для студентов-китаистов (направление 45.03.02 «Лингвистика», профиль «китайский язык»).

Базовой рабочей средой курса выступает OmegaT – свободная кроссплатформенная CAT-система. Она не выполняет машинный перевод сама по себе и работает локально; её задача – организовать работу переводчика и дать доступ к памяти перевода и глоссариям. OmegaT задаёт чёткую структуру проекта и данных: стандартные папки (source/target, glossary и т.д.), ПП, проектные правила сегментации и фильтры форматов – всё это фиксируется на уровне конфигурации и облегчает воспроизводимость. Подключение машинного перевода реализуется через плагины-коннекторы МП, а рутинные операции можно автоматизировать скриптами – поэтому локальный МП-агент добавляется как ещё один источник предложений, а контур принятия решений остаётся у человека [4].

Поскольку OmegaT сама не выполняет перевод, машинный перевод подключается как внешний сервис. В учебных условиях нам важны два требования: работа локально и

умеренные требования к оборудованию. Поэтому в качестве движка мы используем малую языковую модель семейства Qwen 2.5/3 в квантованных сборках (например, q4/q5), что снижает потребление памяти и позволяет запускать систему на типовых ПК [5, 6].

Модель подаётся в OmegaT через локальный сервер с OpenAI-совместимым API (например, llama.cpp или llama-cpp-python), поэтому для пользователя она выглядит как обычный МП-сервис, который можно выбрать в настройках САТ-среды. Такой вариант сохраняет данные на стороне аудитории и даёт возможность управлять версией модели и параметрами генерации в рамках учебного стенда [7, 8].

Чтобы курс не свёлся к набору приёмов, мы связали его с общеевропейскими компетентностными моделями.

1. **Tuning** – подход к проектированию программ «от результатов обучения», где результаты описываются через компетенции (предметные и общие) и служат опорой для содержания, оценивания и качества. Методология широко применялась в Европейском пространстве высшего образования и была адаптирована в РФ в рамках проекта Tuning Russia [9].

2. **DigComp 2.2** – общеевропейская модель цифровой компетентности граждан: 5 областей, 21 компетенция и 8 уровней владения; издание 2.2 добавляет >250 примеров знаний, умений и установок для конструирования заданий и критериев (актуализирует, не меняя ядро из 5 областей) [10].

Опираясь на эти модели, мы выделяем три блока умений: **технологический** – разворачивание стенда, настройка локального сервера, интеграция компонентов, выбор профиля модели и квантизации; **методологический** – поиск и сбор корпусов, очистка и нормализация данных, ведение глоссариев, оформление «паспорта» эксперимента (версии, зерно случайности, параметры); **рефлексивный** – критическая оценка качества машинного перевода, фиксация ограничений и выбор стратегии постредактирования [11]. Эти блоки соотносятся с областями DigComp (данные/контент/решение проблем), поэтому легко ложатся в критерии и рубрикаторы курса. Такое «сшивание» технических и методико-рефлексивных практик соответствует идеи синергии подходов (communicativeness+, culturality+ и др.) как perpetuum mobile развития лингводидактики [3].

Коммуникация и совместная работа в группе студентов организуются через коллективную разметку, согласование терминологии и ведение проектной документации (ППХ-память, глоссарии, журналы изменений). Эти действия напрямую попадают в область 2 модели DigComp «коммуникация и сотрудничество», где акцент делается на обмен данными/контентом и корректные практики атрибуции, – поэтому их удобно превращать в наблюдаемые задания и критерии оценки [3, 10]. Мы сознательно проектируем междисциплинарную среду (язык × данные × инженерные практики), что отражает общую тенденцию к интеграции оснований в лингводидактике [3].

При этом мы фиксируем параметры эксперимента (версию и профиль модели, квантизацию, зерно случайности, наборы данных и разбиения, параметры инференса) и структуру проекта OmegaT. Такая паспортизация вместе с использованием воспроизводимых «подписей» метрик позволяют внешнему читателю воспроизвести опыт и проверить наши выводы на своём стенде.

В учебных аудиториях установка ПО часто ограничена правами администратора, а конфигурации машин заметно различаются. Поэтому мы используем переносимый (portable) стек: набор инструментов, запускающийся без установки с USB-накопителя.

Такой подход позволяет развернуть стенд на любом ПК и воспроизводить задания независимо от локальной политики ИТ-службы; единственным практическим ограничением остается скорость носителя и (если копировать на диск) время переноса. В стек входят программы, у которых есть официальные portable/ZIP-сборки или которые изначально работают без установки: например, WinPython (полноценный Python-дистрибутив «в одной папке»), а также утилиты, которые распространяются в виде самостоятельных исполняемых файлов и не требуют установки – например, FFmpeg.

Для удобства преподавателей и студентов мы собрали этот набор в единый переносимый комплект, который иронично назвали «**флешка судного дня**». Название отражает идею полной автономности: даже при отсутствии интернета и прав администратора стенд сохраняет работоспособность, а все необходимые инструменты находятся под рукой. Версия 0.1 включает базовые программы для перевода, корпусного анализа, мультимедиа и документации. Они подбирались по двум критериям – открытые лицензии и возможность запуска без установки:

- OmegaT – локальная САТ-среда: ПП, глоссарии, подключение внешних МП-сервисов через плагины;
- llama.cpp / llama-cpp-python / KoboldCpp – локальный сервер LLM с OpenAI-совместимым API; обеспечивает единый интерфейс подключения для коннекторов в OmegaT;
- OpenCC – нормализация китайского письма (упрощённое/традиционное, региональные варианты) для проектов zh↔ru;
- AntConc – частотный анализ и конкордансер;
- Calibre – управление коллекциями текстов, конверсия форматов;
- Sublime Text – правка скриптов и разметки;
- GoldenDict – онлайн-оболочка для словарей;
- FFmpeg – консольная обработка аудио/видео (перекодирование, извлечение дорожек);
- Aegisub, Subtitle Edit – тайминг и правка субтитров (мультимодальные задания);
- Shotcut – свободный видеоредактор;
- Audacity – запись и монтаж звука;
- Píper TTS – локальный синтез речи;
- gImageReader + Tesseract – онлайн-OCR для сканов и изображений;
- ScreenToGif – запись экрана для микроотчётов и демонстраций;
- SumatraPDF – лёгкий просмотр PDF/DJVU; доступна portable-сборка;
- XnView MP – просмотр и пакетная обработка изображений; доступна ZIP-версия;
- VLC – кроссплатформенный медиаплеер;
- LibreOffice Portable – офисный пакет без установки (текст/таблицы/слайды);
- Firefox / Vivaldi (portable-варианты) – браузеры для онлайн-демонстраций и локальной документации.

Таким образом, переносимый стек выступает реализацией межпредметной интеграции – соединения инженерных и лингвистических компонентов образовательного процесса [12].

Схема работы предельно прозрачна. Локальный сервер запускается из готового скрипта и предоставляет стандартный HTTP-интерфейс. OmegaT подключается к нему как к внешнему сервису машинного перевода и запрашивает автоматический вариант перевода для каждого сегмента – рядом с совпадениями ПП и терминологией. Для пользователя интерфейс OmegaT не меняется: выбор провайдера и его параметров выполняется в настройках МП-плагина (URL, ключ), а профиль модели, температура и квантизация задаются на стороне локального сервера [4, 7, 8].

Выбор движка МП задаётся аудиторными ограничениями и языковой парой. Мы используем семейство Qwen 2.5/3: у него есть открытые веса и линейка размеров от ~0.5B до 72B, что позволяет подобрать «профиль» под конкретный класс; при этом новое поколение Qwen 3 заявляет поддержку 100+ языков/диалектов, а Qwen стабильно показывает сильные результаты на китайских бенчмарках (напр., C-Eval) – это критично для наших проектов zh↔ru. Портативный запуск обеспечивается квантизацией: использование сборок уровня q4

заметно снижает требования к памяти и даёт приемлемую скорость на обычных ПК; при наличии даже скромного GPU профиль можно поднять [5, 6].

Для пары zh↔ru принципиальны аккуратная нормализация и корректная разметка: мы применяем OpenCC уже на этапе подготовки корпусов и учебных заданий (приведение упрощённого/традиционного письма и региональных вариантов), что заметно снижает шум при сопоставлении сегментов и повышает согласованность терминологии. Далее глоссарии подключаются в OmegaT и начинают задавать предпочтительные соответствия для доменной лексики в том же интерфейсе, где студент видит совпадения ПП и автоматический вариант перевода [4, 13].

Практическая работа строится как последовательность лабораторных шагов. Сначала поднимается локальный сервер модели, предоставляющий стандартный (OpenAI-совместимый) HTTP-интерфейс; это может быть llama.cpp или llama-cpp-python – оба решения документируют запуск сервера одной командой и работу по эндпоинту /v1/chat/completions. Затем сервер подключается в OmegaT как внешний провайдер машинного перевода через диалог «Machine Translation»: указывается локальный URL (и, при необходимости, ключ), после чего автоматический вариант перевода появляется рядом с памятью и глоссариями. На этом же шаге формируется минимальная «базовая линия»: небольшая тестовая выборка переводится без каких-либо адаптаций, чтобы было от чего отталкиваться дальше [4, 7, 8].

Следом организуется работа с данными. Студенты ищут и документируют открытые параллельные источники для zh↔ru и смежных направлений: корпус ООН (UN Parallel Corpus v1.0) как качественный официальный материал и коллекции OPUS (например, WikiMatrix/CCMatrix) как «широкую» базу, требующую описанной фильтрации [14, 15, 16]. Параллельно создаётся собственный «грязный» корпус: из собранных материалов выделяются фрагменты, выполняется сегментация, нормализация через OpenCC и полуавтоматическое выравнивание русских и китайских предложений в нашем интерфейсе. Чтобы не оставаться «в воздухе», мы опираемся на известные постановки задачи и алгоритмы выравнивания предложений (hunalign, Vecalign): даже если инструмент свой, методическая база остаётся проверяемой и воспроизводимой.

После этого показываем, как данные влияют на поведение модели. На основе компактного чистого поднабора запускается краткая адаптация базовой модели (например, в формате лёгкой донастройки), фиксируются версии и параметры, а результаты сравниваются с исходной базовой линией на том же тесте [17, 18]. Смысл здесь не в «метриках ради метрик», а в том, чтобы студент увидел причинную связку между составом корпуса, настройками и качеством результата.

Отдельный учебный эпизод посвящён реальному кейсу «страница → текст → перевод». Китайский фрагмент страницы распознаётся онлайн (gImageReader + Tesseract), затем при необходимости нормализуется через OpenCC, и один и тот же текст переводится тремя способами: напрямую через локальный сервер без системного промпта и с системным промптом (в обоих случаях – на «чистой» и на дообученной модели для сопоставления), а также внутри OmegaT с тем же локальным сервисом, чтобы увидеть поведение в САТ-контуре. Такой триптих наглядно показывает, как меняется результат от контекста вызова и от минимальной адаптации [2, 13].

Завершает цикл аккуратное оценивание и паспортизация. Качество фиксируется sacreBLEU с явной подписью параметров, включая обязательную для китайского настройку токенизации (tokenize=zh); по желанию добавляются языконезависимые метрики, но именно подпись sacreBLEU делает сравнения проверяемыми для внешнего читателя. В «паспорте» каждого проекта кратко перечисляются версия и профиль модели, способ запуска локального сервера, состав и происхождение данных, разбиения на выборки и основные параметры инференса; структура проекта OmegaT (стандартные каталоги и конфигурация) прикладывается как артефакт [4, 19, 20]. Важно подчёркнуть проговорить границы ответственности интерфейсов: выбор размера и квантизации модели осуществляется на

стороне локального сервера (через файл модели/конфигурацию или отдельный эндпоинт), а в OmegaT настраивается подключение провайдера и его параметры – так у читателя не возникнет ложных ожиданий относительно «ползунков» внутри САТ.

Такое линейное развертывание – от нормализации и выравнивания до небольшой донастройки и сопоставимого оценивания – даёт ровно те качества, ради которых мы и строим локальный контур: прозрачность, управляемость и воспроизводимость, при которых выводы курса можно проверить на другой машине и в другой группе, не меняя методологию и артефакты.

Сопоставление с облачными системами проясняет границы применимости подхода. Облако обычно даёт более высокое «среднее» качество и стабильность на поддерживаемых языках и доменах: сравнительные обзоры регулярно показывают, что лидирующие сервисы (DeepL, Google, Microsoft) выигрывают скоростью и зрелостью пайплайна. Педагогическая цена этого удобства – снижение наблюдаемости процесса: путь от данных и настроек к результату для студента остаётся скрытым. Локальная архитектура, напротив, возвращает контроль и позволяет разбирать причинно-следственные связи на собственном стенде; практическим бонусом остаются приватность и предсказуемость обращения с данными в «аудиторном периметре» [5, 21].

Последовательность небольших, но наблюдаемых шагов формирует у будущего переводчика исследовательскую привычку: смотреть на текст, корпус и инструмент как на единую систему, где каждое решение оставляет след в данных и итоговом варианте. Когда эта привычка укореняется, модель перестаёт быть центром внимания курса: на первый план выходят дизайн задания, качество и прозрачность корпуса, договорённости о терминологии и культура постредактирования. Именно здесь локальный стенд оказывается дидактически продуктивным: он делает управление параметрами и интерпретацию метрик предметом обсуждения, а не фоном.

В результате онлайн-контур локального МП-агента в связке с OmegaT демонстрирует, как технические решения становятся прямым продолжением педагогических требований: от «чёрного ящика» – к управляемой и воспроизводимой практике. Для занятий разумно стартовать с «малого» профиля семейства Qwen: на сегодня это либо Qwen 3 (открытые веса под Apache 2.0, мультиязычность, сильные результаты на китайских бенчмарках), либо Qwen 2.5 в вариантах с открытой лицензией. В качестве сравнимой внешней точки – держать под рукой NLLB-200 для онлайн-инфераенса (через CTranslate2) и фиксировать различия на одном и том же тест-наборе [5, 6, 22, 23]. Такой «минимальный» набор даёт честный диапазон: от контролируемого локального сценария до сильного открытого baseline [24].

План расширения очевиден: добавить на носитель готовые профили под несколько размеров Qwen 3 и совместимые конфигурации локального сервера (OpenAI-совместимые эндпоинты для быстрого подключения в OmegaT), включить преднастроенный пакет NLLB-200/CTranslate2 для онлайн-сравнений, а также шаблоны «паспорта проекта» и карточек датасетов (с контрольными суммами и версиями). Это усилит переносимость стенда между аудиториями и семестрами.

### *Библиографический список*

1. Тарева, Е. Г. Цифровая эпоха и педагогические профессии / Е. Г. Тарева // Вестник МГПУ. Серия «Философские науки». – 2018. – С. 85–90.
2. Suleimanova, O. A. Towards synergistic combination of traditional and innovative digital teaching and research practices / O. A. Suleimanova // Training, Language and Culture. – 2020. – Vol. 4, No. 4. – P. 39–50.
3. Тарева, Е. Г., Тарев, Б. В., Савкина, Е. А. Полиподходность и междисциплинарность – *perpetuum mobile* развития лингводидактики / Е. Г. Тарева, Б. В. Тарев, Е. А. Савкина // Язык и культура. – 2022. – № 57. – С. 274–291.

4. OmegaT. User Manual. – URL: официальный сайт OmegaT (дата обращения: 14.10.2025).
5. Xu J., et al. On-Device Language Models: A Comprehensive Review. –2024. – arXiv:2409.00088. – URL: arXiv (дата обращения: 14.10.2025).
6. Qwen2.5-7B-Instruct: карточка модели (Apache-2.0). – URL: Hugging Face (дата обращения: 14.10.2025).
7. llama.cpp: репозиторий проекта (OpenAI-совместимый HTTP-сервер). – URL: GitHub (дата обращения: 14.10.2025).
8. llama-cpp-python: OpenAI-совместимый веб-сервер. – URL: документация (дата обращения: 14.10.2025).
9. Tuning Educational Structures in Europe. Final Report. – Bilbao: University of Deusto, 2009. – 309 с. – URL: Tuning Academy (дата обращения: 14.10.2025).
10. Вуорикари Р., и др. DigComp 2.2: The Digital Competence Framework for Citizens. – Luxembourg: Publications Office of the EU, 2022. – 188 с. – URL: репозиторий Европейской комиссии (дата обращения: 14.10.2025).
11. Сулейманова, О. А.; Нерсесова, Э. В.; Вишневская, Е. М. Технологический аспект подготовки современного переводчика / О. А. Сулейманова, Э. В. Нерсесова, Е. М. Вишневская // Филологические науки. Вопросы теории и практики. – 2019. – Т. 12, № 7. – С. 313–317.
12. Гулиянц, А. Б.; Гулиянц, С. Б. Применение интегративного подхода в профессиональной подготовке переводчиков в вузе / А. Б. Гулиянц, С. Б. Гулиянц // Филологические науки. Вопросы теории и практики. – 2022. – Т. 15, № 11. – С. 72–74.
13. Open Chinese Convert (OpenCC): репозиторий проекта. – URL: GitHub (дата обращения: 14.10.2025).
14. Ziemska M., et al. The United Nations Parallel Corpus v1.0. – 2016. – URL: ACL Anthology / UN (дата обращения: 14.10.2025).
15. Tiedemann J. OPUS – The Open Parallel Corpus. – URL: OPUS (каталог корпусов) (дата обращения: 14.10.2025).
16. Schwenk H., et al. WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs. – 2021. – URL: ACL Anthology / OPUS (дата обращения: 14.10.2025).
17. Hu E. J., et al. LoRA: Low-Rank Adaptation of Large Language Models. – 2021. – URL: arXiv:2106.09685 / OpenReview (дата обращения: 14.10.2025).
18. Detmers T., et al. QLoRA: Efficient Finetuning of Quantized LLMs. – 2023. – URL: arXiv:2305.14314 / ACM DL / GitHub (дата обращения: 14.10.2025).
19. Post M. A Call for Clarity in Reporting BLEU Scores (sacreBLEU). – 2018. – arXiv:1804.08771. – URL: arXiv (дата обращения: 14.10.2025).
20. No Language Left Behind (FLORES-200). – 2022. – arXiv:2207.04672. – URL: arXiv / Hugging Face (дата обращения: 14.10.2025).
21. Qu G., et al. Mobile Edge Intelligence for Large Language Models: A Contemporary Survey. – 2024. – URL: ПредпUBL. / TechRxiv (дата обращения: 14.10.2025).
22. CTranslate2: быстрый движок инференса для Transformer-моделей. – URL: GitHub (дата обращения: 14.10.2025).
23. NLLB-200 в CTranslate2: руководство/примеры. – URL: OpenNLP Forum / HF (дата обращения: 14.10.2025).
24. Гераймович, Е. О., Коптев, Д. А., Любашев, Н. А., Рожковская, Э. Д., Шаренкин, Д. Р., Шипунов, М. А. Практическое использование цифровых технологий в проведении лингвистических исследований: учебно-методическое пособие / Е. О. Гераймович, Д. А. Коптев, Н. А. Любашев, Э. Д. Рожковская, Д. Р. Шаренкин, М. А. Шипунов. – Москва : Языки Народов Мира, 2024. – 111 с.