

Функциональные характеристики

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

YasnoCore LLM 1.0.0 - локальная программная надстройка для управляемого использования и адаптации сторонних языковых моделей. Собственный программный код реализует шлюз, контроль входных запросов, подготовку наборов данных, оценку результатов, контроль релиза и запуск адаптерного обучения.

Предоставляемый экземпляр содержит 11 Python-модулей. Их содержимое совпадает с зафиксированным исходным текстом, поданным в ФИПС 05.10.2026. Базовые веса модели, llama.cpp, PyTorch и другие сторонние продукты не заявляются как собственная разработка ООО «ЯСНОКОР».

Функция	Реализация версии 1.0.0
Локальный шлюз	/health, /v1/models, /v1/chat/completions, /v1/yasnocore/analyze
Контроль запросов	Проверка структуры JSON, ролей сообщений, размера тела и длины текста
Подготовка данных	Проверка обязательных полей и основания прав; отклонение дубликатов; train/validation с фиксированным seed
Оценка	Нормализованное совпадение, WER, CER и точность JSON-полей
Контроль релиза	Перечень файлов и SHA-256
Обучение	Синтетический LoRA smoke-test; отдельный запуск train-adapter в подходящем окружении

Заявляемая версия не является самостоятельно обученной базовой языковой моделью. Качество генерации, языковой охват и вычислительные ресурсы зависят от выбранной модели и данных. Готовность работы на десяти тюркских языках этим комплектом не подтверждается.

Границы продукта и интерфейсы

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

Программное ядро

Серверный API работает только на loopback-адресе. Команды CLI: serve, doctor, prepare-dataset, evaluate, manifest, smoke-train и train-adapter. Для шлюза и основных утилит достаточно Python 3.11; обязательных сторонних pip-зависимостей у ядра нет.

В записи набора данных требуются instruction, response, language, source_id и rights_basis. Версия 1.0.0 перемешивает строки с фиксированным seed и разделяет их на train/validation. Дополнительную изоляцию связанных источников и отдельную test-выборку организует оператор до подготовки данных; поведение последующей ветки 1.1 сюда не переносится.

Дополнительный веб-кабинет

Сайт <https://llm.yasnosoft.ru/> обслуживает тот же кабинет, что llm.yasnocore.ru. Он является отдельным дополнительным компонентом подготовки корпуса: личный вход, роли партнёра/языкового эксперта/администратора, проверка текста и отдельный допуск по правам, выгрузка JSONL. Для установки Python-ядра кабинет не требуется и в его архив не входит.

Заявляемая конфигурация

Astra Linux Special Edition 1.8.2, «Воронеж», Python 3.11; локальный inference-бэкенд на 127.0.0.1:8080. Полноценное обучение исходного checkpoint через Transformers/PEFT и GPU, внешние облачные API и лабораторные модели не включены в проверенный профиль CPU-сервера.

ООО «ЯСНОКОР», ИНН 2100033229, ОГРН 1262100003175. Поддержка: info@yasnocore.ru, +7 (927) 841-00-61. Регистрационный документ Роспатента ожидается; в Минцифры этот комплект пока не направляется.

Инструкция установки на Astra Linux

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

1. Среда и условия

Проверено: Astra Linux SE 1.8.2 «Воронеж», Python 3.11.2, x86-64. Профиль защиты и замкнутая программная среда сохраняются включёнными. Серверный шлюз и CLI не требуют Windows Server, IIS, облачного сервиса или Composer.

Для собственного ядра ориентир - 1 vCPU, 1 ГБ ОЗУ и 1 ГБ диска без модели, корпусов и архивов. Это ориентир для небольшого экземпляра, не нагрузочная гарантия. Ресурсы inference-модели рассчитываются отдельно. На рабочем общем сервере сейчас 16 vCPU и около 32 ГБ ОЗУ, без GPU.

2. Состав поставки и проверка

Архив YasnoCore_LLM_1.0.0_Astra_20261019.zip содержит исходный текст, тесты, синтетический пример, установщик и README. Веса моделей, реальные корпуса, рабочие конфигурации, учётные записи и ключи отсутствуют. Распакуйте его в служебный каталог вне web root и перейдите в YasnoCore_LLM_1.0.0_Astra.

```
sha256sum -c SHA256SUMS.txt
sudo apt-get install python3-venv python3-setuptools \
python3-wheel curl
```

3. Локальный inference-бэкенд

Администратор отдельно устанавливает лицензированную модель и совместимый inference-сервер. Бэкенд должен слушать только 127.0.0.1:8080 и поддерживать /health, /v1/models и /v1/chat/completions. В настройке ядра допускается только локальный HTTP endpoint.

```
curl --fail http://127.0.0.1:8080/health
curl --fail http://127.0.0.1:8080/v1/models
```

До распространения сторонней модели проверьте лицензию точного checkpoint. Нативные исполняемые файлы под ЗПС должны иметь доверенную подпись по правилам Astra. Не отключайте защиту для установки неподписанного бинарного файла. Приватные подписывающие ключи в поставку не входят.

Установка службы и контроль

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

4. Новая установка

Замените YOUR_MODEL_ID идентификатором, который выдаёт локальный /v1/models.

Выполните из каталога пакета:

```
sudo env YASNOCORE_LLM_MODEL=YOUR_MODEL_ID \  
bash deploy/install_astra.sh
```

Установщик создаёт /opt/yasnocore-llm, отдельного системного пользователя yasnocore-llm, venv, runtime.env и службу yasnocore-llm. При существующем каталоге или unit-файле останавливается. Пакет устанавливается локально без загрузки рп-зависимостей. Код и venv принадлежат root, служебной группе предоставлены чтение и исполнение; запись службы ограничена её журналом.

Сервис слушает 127.0.0.1:8090. В runtime.env записываются локальный адрес бэкенда, alias модели, таймаут и путь журнала. Не публикуйте этот порт и не добавляйте к нему открытый reverse проху: в версии 1.0.0 нет сетевой пользовательской аутентификации.

```
systemctl status yasnocore-llm --no-pager  
curl --fail http://127.0.0.1:8090/health  
/opt/yasnocore-llm/venv/bin/yasnocore-llm doctor  
/opt/yasnocore-llm/venv/bin/python -m unittest discover \  
-s /opt/yasnocore-llm/app/tests -v
```

5. Проверочный запрос

```
curl --fail -H 'Content-Type: application/json' \  
--data '{"messages":[{"role":"user","content":"Привет"}], "max_tokens":16}' \  
http://127.0.0.1:8090/v1/chat/completions
```

Запрос выполняется с самого сервера. Проверьте HTTP 200 и непустой choices[0].message.content. Ответ модели не является проверкой фактической правильности и должен оцениваться пользователем.

CLI, обучение и обслуживание

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

6. Подготовка синтетических данных

```
CLI=/opt/yasnocore-llm/venv/bin/yasnocore-llm
$CLI prepare-dataset \
/opt/yasnocore-llm/app/examples/regional_sample.jsonl \
/var/lib/yasnocore-llm/prepared-demo
```

Используйте каталог, доступный оператору на запись. Для служебного пользователя запускайте команду через runuser. Проверьте manifest.json, train.jsonl и validation.jsonl. Оригиналы данных храните отдельно; реальные материалы допускаются только при подтверждённых правах и согласованных целях обработки.

7. Синтетический LoRA smoke-test

```
sudo apt-get install python3-torch python3-numpy
/opt/yasnocore-llm/venv/bin/yasnocore-llm smoke-train
```

Этот короткий тест использует случайные синтетические матрицы и не меняет рабочие веса Qwen. Полное train-adapter требует отдельного проверенного окружения Transformers/PEFT и исходного checkpoint; GGUF для inference нельзя напрямую использовать как такой checkpoint. Установка и обучение внешней модели в данный установщик не включены.

8. Обновление, резервирование, удаление

До обновления сохраните код, runtime.env, конфигурацию службы и необходимые данные. Проверяйте новую версию на отдельном стенде. Этот установщик поверх существующего экземпляра не запускается. Для снятия стенда остановите и отключите только его службу, сохраните нужные результаты и архивируйте каталог. Удаление рабочих данных выполняется отдельно по решению владельца.

Веб-кабинет опубликован по HTTPS на двух доменах, но это не сетевой вход к API ядра. Для интеграции клиентских приложений требуется отдельный аутентифицированный защищённый шлюз.

SHA-256 установочного ZIP:

```
1e7e5743b3d4a2e0db287d794cc5caff
4f8d6a51a4a08b466beb79db3b16d6ad
```

Руководство эксплуатации

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

Локальный API

Метод и маршрут	Назначение
GET /health	Состояние ядра, версия и доступность бэкенда
GET /v1/models	Идентификаторы моделей от бэкенда
POST /v1/chat/completions	Передача проверенных сообщений локальной модели
POST /v1/yasnocore/analyze	Формирование задания анализа с JSON-ответом

Для chat передайте объект messages со списком ролей system/user/assistant и непустым текстом; поддерживаемые параметры генерации передаются бэкенду. Шлюз использует настроенный alias модели и выключает потоковый режим. Некорректный JSON или роль отклоняются с HTTP 400; недоступный бэкенд возвращает 503.

В analyze передаётся text и дополнительные поля, поддерживаемые build_analysis_messages. Версия 1.0.0 формирует собственный запрос со stream=false и JSON response_format; произвольные параметры генерации из входного analyze не следует считать автоматически переданными модели. Для короткой ограниченной генерации используйте chat с max_tokens.

Журналирование

Для POST сохраняются request_id, маршрут, HTTP-статус, длительность, SHA-256 входного объекта и локальный адрес клиента. Сам входной текст штатный аудит ядра не сохраняет. Веб-кабинет и inference-бэкенд могут иметь отдельные журналы; их правила хранения устанавливает администратор.

Граница доступа

Оператор работает на сервере либо через отдельный согласованный контур. Ядро 1.0.0 намеренно ограничено loopback; пользовательские роли веб-кабинета не являются авторизацией API. Не открывайте 8080/8090 наружу.

Утилиты и дополнительный кабинет

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

Утилиты

doctor проверяет бэкенд и модели. **prepare-dataset** создаёт train/validation и manifest с контрольными суммами. **evaluate** читает JSONL с expected/actual и считает метрики. **manifest** формирует опись релиза. **smoke-train** выполняет синтетическое упражнение LoRA; **train-adapter** требует отдельного окружения и checkpoint.

Каждая строка входного набора должна содержать instruction, response, language, source_id и rights_basis. Проверьте права, происхождение и группировку источников до подготовки. Значения прав unknown/неизвестно/нет и дублирующиеся пары отклоняются. Укажите отдельный выходной каталог, сохраняйте оригинал.

Кабинет подготовки корпуса

На <https://llm.yasnosoft.ru/?page=login> партнёр входит по индивидуальному доступу. Партнёр добавляет текст и перевод; языковой эксперт проверяет содержание; администратор отдельно проверяет основание прав и допуск к выгрузке. Веб-часть использует свою MariaDB и серверные сессии. Материалы не становятся публичными от их загрузки.

Первый вход может требовать смены временного пароля. После редактирования материала допуски необходимо получить заново. Подготовка/выгрузка корпуса сама по себе не запускает обучение модели. Аудио, видео и PDF не обрабатываются Python-ядром LLM как распознавание речи.

Неисправности и поддержка

HTTP 400: проверьте JSON/поли/лимиты; 503: проверьте локальный inference; нет подключения: systemctl status и journalctl -u yasnocore-llm. Веб-кабинет при отсутствии входа переводит на форму авторизации. Не отключайте защиту и не публикуйте порт для диагностики.

Поддержка: info@yasnocore.ru, +7 (927) 841-00-61. Сообщите версию, время, команду и обезличенный фрагмент ошибки. Не передавайте пароли, корпуса и рабочие документы в обычное обращение.

Техническая архитектура и безопасность

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

Компонент	Граница
Собственное ядро	11 модулей Python, версия 1.0.0, без обязательных pip-зависимостей
Gateway	127.0.0.1:8090; контроль формата/лимитов и аудит
Inference	Отдельный локальный процесс 127.0.0.1:8080; модель и нативный сервер лицензируются отдельно
CLI	Подготовка наборов, метрики, опись, smoke-test и запуск обучения
Дополнительный портал	Apache/PHP/MariaDB, HTTPS, аутентификация и роли подготовки корпуса

Путь запроса: локальное клиентское приложение → YasnoCore LLM gateway → локальный inference-бэкенд → ответ клиенту. При такой конфигурации пользовательский текст не отправляется в облачный AI-сервис. Автоматического обращения к облачному API в собственном ядре нет.

Два домена llm.yasnocore.ru и llm.yasnosoft.ru используют один DocumentRoot и одну конфигурацию кабинета. Новый домен не создаёт вторую базу данных. Сессии браузера привязаны к имени хоста; при переходе между доменами может понадобиться отдельный вход.

Защита исполнения

Проверенная ОС - Astra Linux SE 1.8.2 «Воронеж», режим 1. Служба работает от отдельной непривилегированной записи. Код принадлежит root; запись разрешена в выделенные служебные каталоги. Профиль защиты, ЗПС и МКЦ не отключаются. Нативный inference должен соответствовать правилам доверенной подписи Astra.

Сторонние компоненты и ограничения

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

Компонент	Роль / поставка
CPython 3.11	Интерпретатор; отдельно, PSF License
llama.cpp	Внешний inference-сервер; отдельно, MIT
Веса модели	Настраиваются отдельно; лицензия точного checkpoint проверяется до распространения
PyTorch / NumPy	Опциональный CPU smoke-test; пакеты Astra отдельно
Transformers / PEFT	Опциональное полное обучение; не входят в подтверждённый CPU-профиль
Apache / PHP / MariaDB	Среда отдельного кабинета; не обязательны для Python-ядра

В проверенном окружении бэкэнд имеет рабочий alias Qwen3.8-27B. Alias - конфигурационное имя сервера; оно не является доказательством происхождения, версии upstream или лицензионных прав на checkpoint. Веса и лабораторные модели в архив не входят; их права не присваиваются ЯСНОКОР.

Наборы НТРК, частные тексты, чувашско-русские корпуса и адаптеры рабочих исследований не публикуются в документации и не передаются с проверочным экземпляром. Для использования каждого источника нужно отдельное документированное основание.

Внутренние API не имеют внешней пользовательской аутентификации. Защита строится на loopback и изоляции сервера; для внешних пользователей нужен отдельный защищённый шлюз. Эти документы не подтверждают формальную оценку защиты, работу с ненулевыми метками/МРД, качество на десяти языках либо уже выданный сертификат Ready for Astra.

Жизненный цикл и сопровождение

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

Правообладатель и разработчик программной надстройки - ООО «ЯСНОКОР», ИНН 2100033229, ОГРН 1262100003175. Обращения: info@yasnocore.ru, +7 (927) 841-00-61. Сроки и объем сопровождения устанавливаются отдельным договором.

Выпуск и идентификация

Рабочий код и исходный текст для регистрации фиксируются отдельными архивами с SHA-256. Депонированный снимок 1.0.0 не переписывается при развитии ветки 1.1. В установочную поставку входят проверяемые собственные модули; веса и сторонние корпуса учитываются отдельно.

Устранение неисправностей

Обращение регистрируется, воспроизводится на отдельном стенде, оценивается влияние на доступность и данные. Исправление проходит тесты затронутых функций и проверку в целевой ОС. До обновления сохраняются предыдущий код, настройки, службы и необходимые данные. При неуспехе выполняется согласованный откат.

Персонал и эксплуатационные обязанности

Нужны компетенции Python/HTTP/JSON, системного администратора Astra Linux/systemd и, для кабинета, PHP/MariaDB. Владелец данных определяет допуск, сроки хранения, резервирование и цели обучения; языковую правильность проверяет уполномоченный эксперт. Количество сотрудников и SLA определяются масштабом внедрения и договором.

Развитие и завершение

Новые языки и адаптеры вводятся после проверки прав, данных и отдельного качества. Успешный smoke-test не означает обучения рабочей модели. При прекращении использования сервис останавливается, необходимые данные выгружаются владельцу, доступ отзывается; уничтожение выполняется по его решению.

Информация о стоимости

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

YasnoCore LLM предоставляется по отдельному лицензионному договору. Фиксированный публичный тариф в настоящем комплекте не устанавливается; стоимость рассчитывается индивидуально по запросу на info@yasnocore.ru. Документ не является публичной офертой.

Основа расчёта

Позиция	Что определяется в предложении
Простая лицензия	Срок, количество экземпляров/контуров, разрешённое использование
Развёртывание	Среда, количество серверов, настройка и приёмка
Сопровождение	Период, объём работ, режим поддержки
Адаптация	Согласованный набор задач, данные и критерии качества
Вычисления	Предоставляются заказчиком либо рассчитываются отдельно

Цена предложения указывается в рублях. Применимость НДС и налоговая ставка определяются на дату предложения и отражаются в договоре; произвольная ставка или безусловное освобождение этим документом не заявляются.

Права на сторонние модели, inference-сервер и инфраструктурное ПО регулируются их отдельными лицензиями. Наличие доступа к веб-странице не означает выдачи лицензии или права распространять исходный текст, веса и корпуса.

Для подготовки расчёта сообщите задачи, модель использования, инфраструктуру, предполагаемые объёмы и срок. Платные услуги, облачная подписка и обязательство приобретать сервер автоматически не возникают.

Предварительная техническая проверка

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

19.10.2026 проверен устанавливаемый экземпляр YasnoCore LLM 1.0.0 на Astra Linux SE 1.8.2 «Воронеж». Это отчёт разработчика, а не официальный протокол САП и не свидетельство уже выданной совместимости.

Проверка	Результат
Идентичность исходников	11 модулей совпадают с поданным в ФИПС снимком
Чистая установка	Отдельный каталог, пользователь, systemd-сервис и порт 18090; успешно
Штатные unit-тесты	10 из 10 успешно
API health / models	HTTP 200
Некорректная роль / JSON	HTTP 400
Неизвестный маршрут	HTTP 404
Локальный inference	HTTP 200, непустой ответ, max_tokens=16
Новые и прежние HTTPS	Оба домена HTTP 200 с доверенным сертификатом
Закрытый кабинет	Без входа HTTP 303 на login

Основной шлюз и inference остались на 127.0.0.1:8090/8080. Временный стенд после испытания остановлен и архивирован непублично. Рабочие модели и реальные корпуса для тестового обучения не использовались.

Синтетическое обучение и границы результата

YasnoCore LLM 1.0.0 | ООО «ЯСНОКОР» | редакция 19.10.2026

LoRA smoke-test

На подписанном системном PyTorch 1.13.0a0 выполнено 20 шагов на случайных синтетических матрицах. Обучаемых параметров: 256. Loss до: 0.01009477; после: 0.00778197. Это проверяет выполнение кода обучения и не меняет рабочие веса базовой модели.

Исправление установщика

При первой чистой установке выявлен недостаточный групповой доступ выделенной службы к новому `venv`. Исправлены права только установочного окружения: `root` сохраняет владение, служебная группа получает чтение/исполнение. Итоговый ZIP повторно установлен с нуля; проверка службы и API успешна. Код 11 депонированных Python-модулей не менялся.

Идентификация итоговой поставки

Файл: YasnoCore_LLM_1.0.0_Astra_20261019.zip. 19 файлов; без рабочей конфигурации, весов, корпусов, паролей и ключей. SHA-256:

```
1e7e5743b3d4a2e0db287d794cc5caff
4f8d6a51a4a08b466beb79db3b16d6ad
```

Что не проверялось

Не выполнены нагрузочные, длительные, отказоустойчивые испытания, полный аудит безопасности, обучение реального checkpoint через Transformers/PEFT, качество на десяти языках, МРД и ненулевые метки. Лицензия точного upstream checkpoint не подтверждается его конфигурационным `alias`; веса не включены в поставку.

Проверка выполнена на общем CPU-сервере 16 vCPU / около 32 ГБ ОЗУ. Результаты не устанавливают SLA или минимальную скорость генерации. Официальные методику и объём совместимости определяет Astra в отдельной заявке.