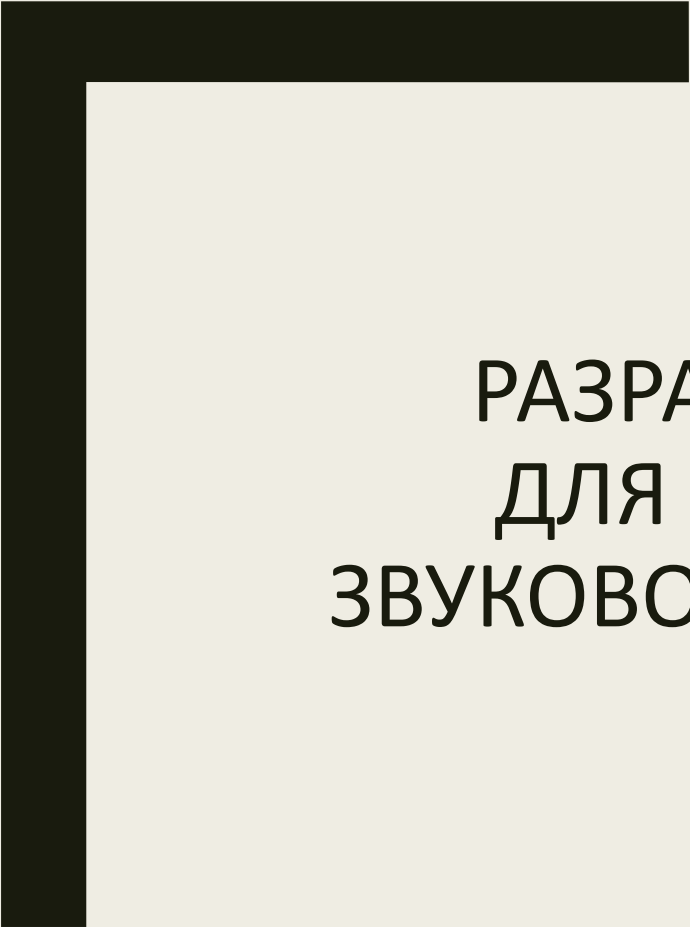




РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ПЕРЕВОДА ТЕКСТА И РЕЧИ ЗВУКОВОГО ЯЗЫКА В ЖЕСТОВЫЙ ЯЗЫК

Авторы: студенты АлтГУ
Меньшиков Сергей и Казаков Геннадий

Введение и актуальность

- Субтитры и образовательные материалы для РЖЯ
- Промежуточный шаг для синтеза жестов
- Низкоресурсность РЖЯ делает решения на основе правил прагматичными для MVP



Лингвистические особенности РЖЯ

- SOV (Subject - Object - Verb)
Наиболее употребим.
- Опускание служебных слов
Предлоги, связки, вспомогательные глаголы часто отсутствуют.
- Немануальные маркеры
Мимика/кивки кодируют вопрос, отрицание, специфику.
- Выражение числа
Повторение жеста или классификаторы.

Архитектура: модуль предобработки и морфологического анализа

“Мама учит сына по красивой книжке”

```
1 {'text': 'Мама', 'lemma': 'мама', 'upos': 'NOUN', 'number': 'Sing'}  
2 {'text': 'сына', 'lemma': 'сын', 'upos': 'NOUN', 'number': 'Sing'}  
3 {'text': 'книжке', 'lemma': 'книжка', 'upos': 'NOUN', 'number': 'Sing'}  
4 {'text': 'красивой', 'lemma': 'красивый', 'upos': 'ADJ'}  
5 {'text': 'учит', 'lemma': 'учить', 'upos': 'VERB'}
```

Модуль лексического маппинга

“Мама учит сына по красивой книжке”

```
1 {'text': 'Мама', 'lemma': 'мама', 'upos': 'NOUN', 'number': 'Sing', 'lemma_dict': 'мама'}
2 {'text': 'сына', 'lemma': 'сын', 'upos': 'NOUN', 'number': 'Sing', 'lemma_dict': 'сын'}
3 {'text': 'книжке', 'lemma': 'книжка', 'upos': 'NOUN', 'number': 'Sing', 'lemma_dict':
  'книга'}
4 {'text': 'красивой', 'lemma': 'красивый', 'upos': 'ADJ', 'lemma_dict': 'красивый'}
5 {'text': 'учит', 'lemma': 'учить', 'upos': 'VERB', 'lemma_dict': 'учить'}
```

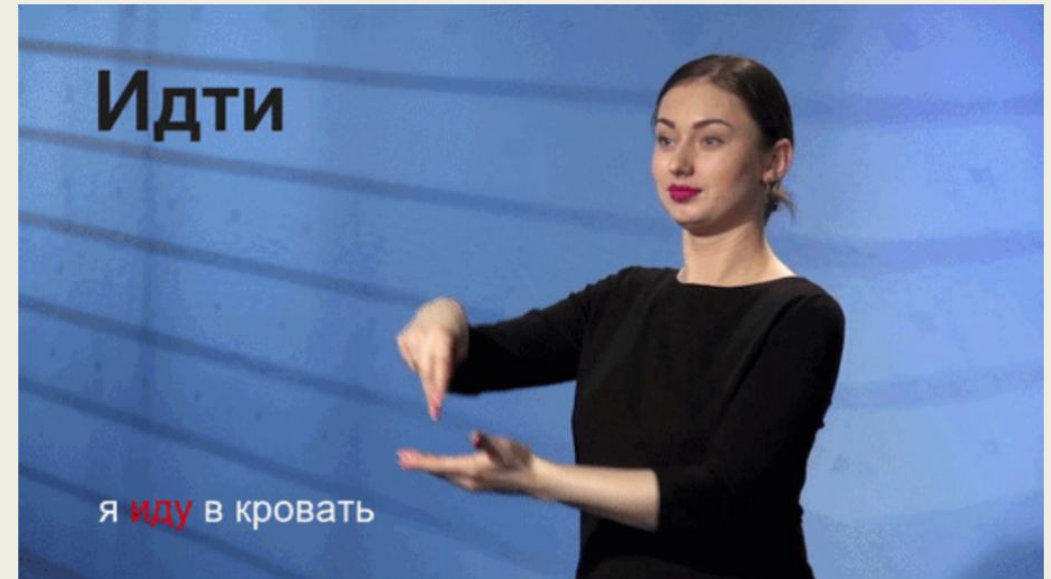
Модуль разрешения многозначности (WSD)

“Мама учит сына по красивой книжке”

```
1 {'text': 'Мама', 'lemma': 'мама', 'upos': 'NOUN', 'number': 'Sing', 'lemma_dict': 'мама',  
  'gloss': 'МАМА'}  
2 {'text': 'сына', 'lemma': 'сын', 'upos': 'NOUN', 'number': 'Sing', 'lemma_dict': 'сын',  
  'gloss': 'СЫН'}  
3 {'text': 'книжке', 'lemma': 'книжка', 'upos': 'NOUN', 'number': 'Sing', 'lemma_dict':  
  'книга', 'gloss': 'КНИГА'}  
4 {'text': 'красивой', 'lemma': 'красивый', 'upos': 'ADJ', 'lemma_dict': 'красивый',  
  'gloss': 'КРАСИВЫЙ'}  
5 {'text': 'учит', 'lemma': 'учить', 'upos': 'VERB', 'lemma_dict': 'учить', 'gloss':  
  'УЧИТЬ_ДРУГОГО'}
```

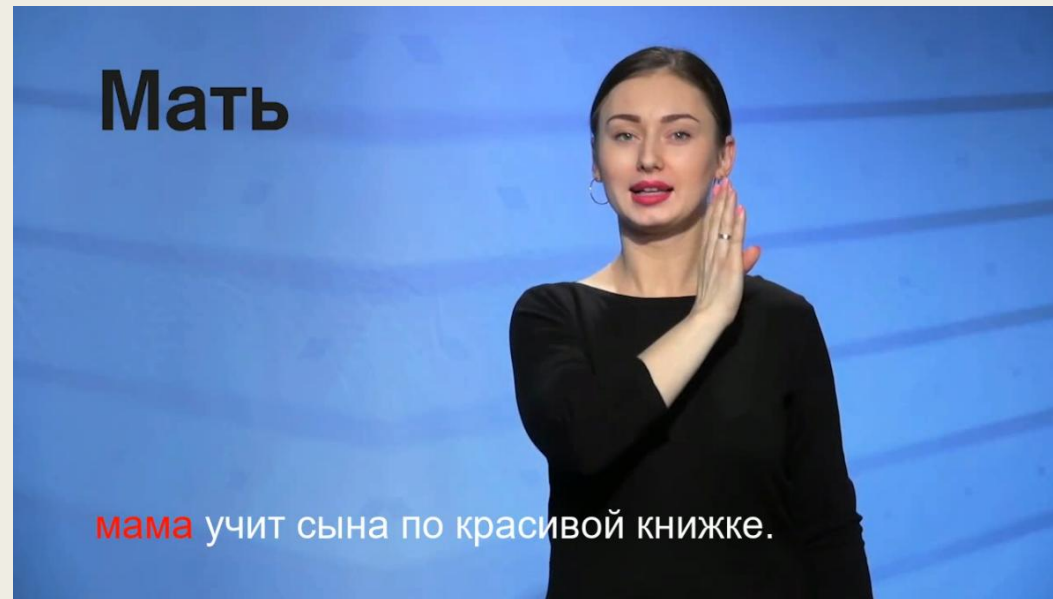
Модуль изменения порядка слов

- Гибкий порядок слов в РЖЯ
- Структура SOV
- Алгоритм:
 1. Токенизация и синтаксический анализ.
 2. Классификация токенов по ролям: части речи и синтаксические роли.
 3. Привязка прилагательных к соответствующим существительным.
 4. Сборка SOV-последовательности:
 $S \rightarrow \text{Mod} \rightarrow O \rightarrow \text{Obl} \rightarrow V$
с добавлением прилагательных после существительных.



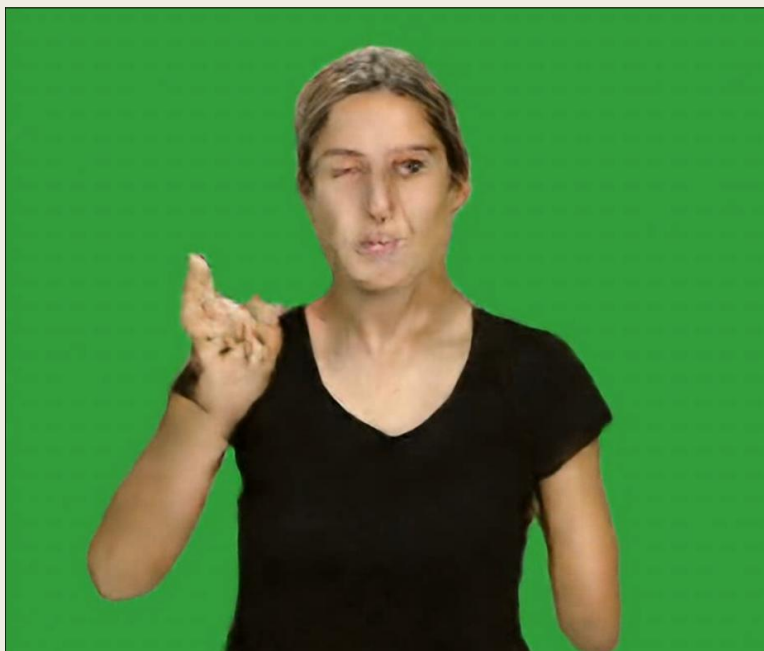
Модуль визуализации

Собирает готовый видеоролик для демонстрации работы модулей из заготовленных видео жестов.



Модуль для немецкого языка (DGS)

Интегрирован сторонний проект spoken-to-signed-translation.



Как ИИ помогает делать анализ предложений?

Проблема: точный синтаксис и морфология нужны для SOV.

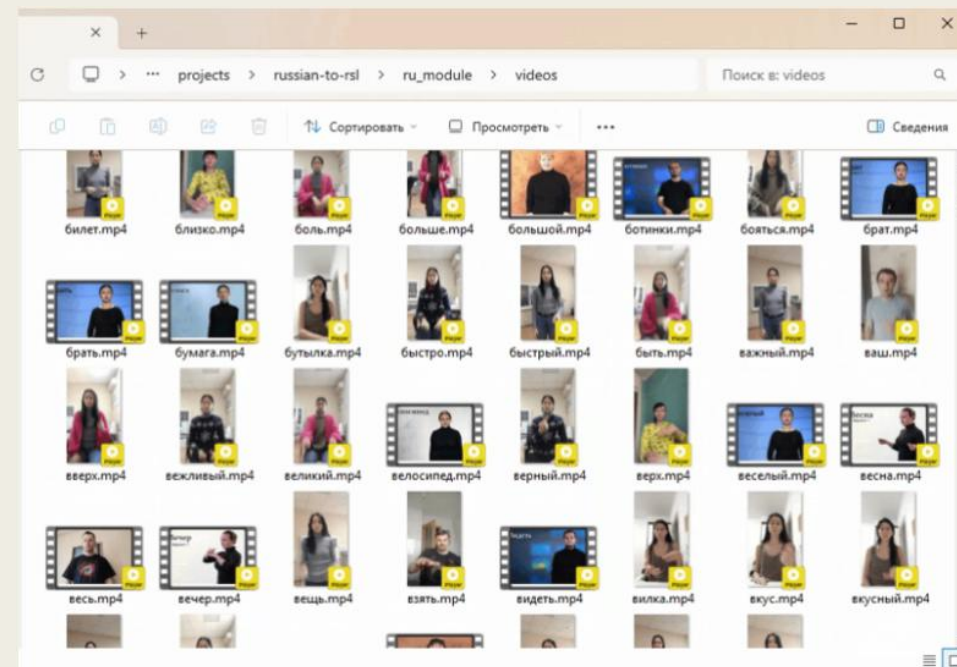
Что пробовали: `rumorphy3`, `natasha` — недостаточно точны.

Что выбрали: **stanza** (нейросетевая).

Почему: лучше всех определяет части речи и синтаксические зависимости.

Словарь

- 100 -> ~500 глосс.
- Самые часто используемые слова русского языка.
- Источники из открытого доступа:
ARK MEDIA,
SpreadTheSign,
SLOVO



Как ИИ помогает выбирать верную глоссу по контексту?

Проблема: одному русскому слову — несколько возможных глосс.

Пример: «Учит» -> «УЧИТЬ_СЕБЯ» (учится сам) или «УЧИТЬ_ДРУГОГО» (учит кого-то).

Решение: используем **ruBERT** — нейросеть, обученную на русском языке.

Как работает: ruBERT строит вектор смысла слова из контекста -> сравниваем с эталонами из словаря -> выбираем подходящую глоссу.

Итог: контекст, а не просто словарное значение.

Разработка веб-приложения: выбор технологий

- HTML + CSS + JavaScript: фронтенд
- FastAPI (Python): бэкенд



HTML



CSS



JS

Распознавание голоса

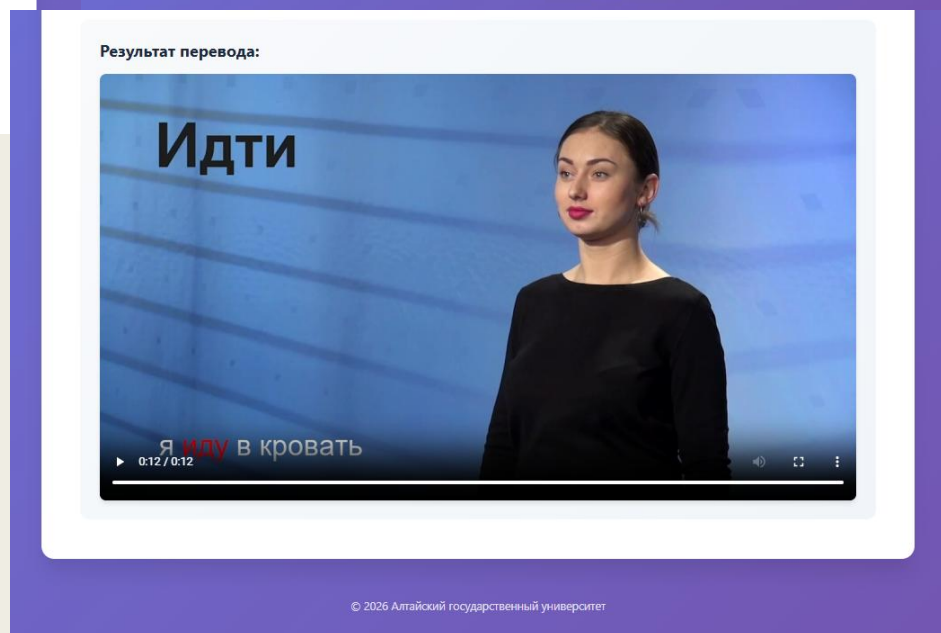
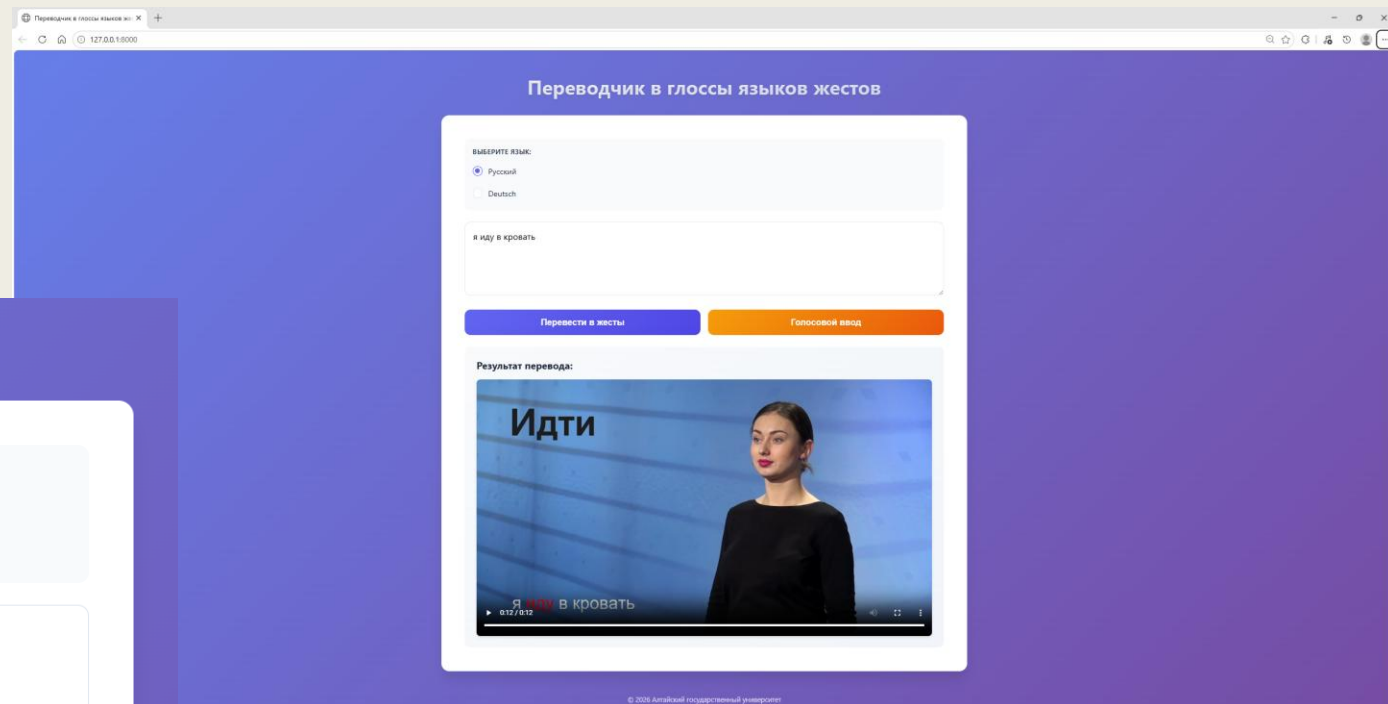
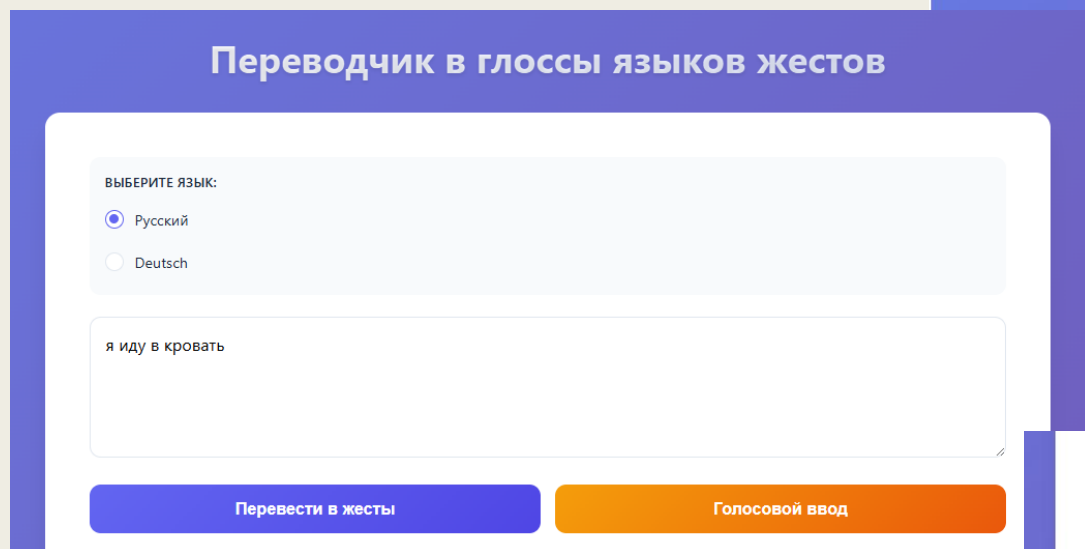
- MediaRecorderAPI – встроенное средство JS
Записываем аудио, конвертируем webm в wav.
- Vosk – офлайн модели распознавания голоса
Распознаём голос, возвращаем JSON.



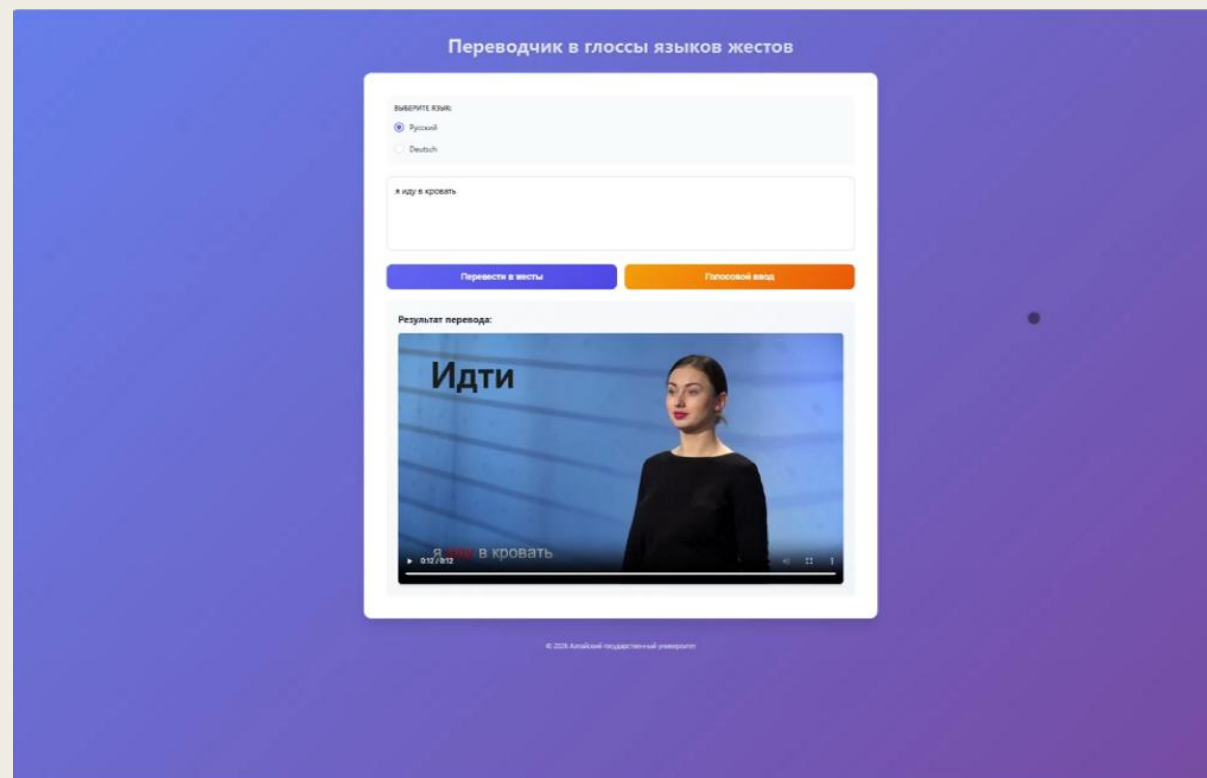
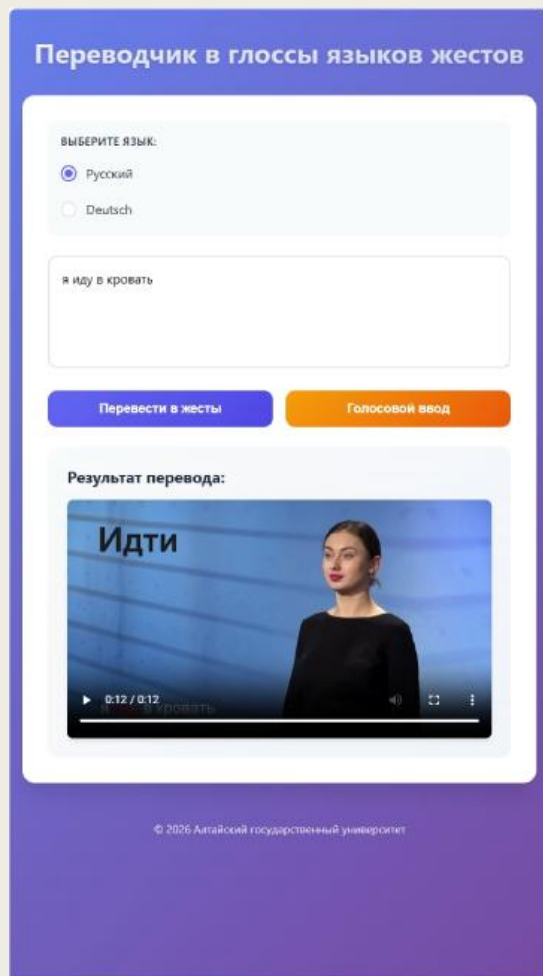
Интеграция модулей

- 1. РЖЯ: была написана функция-обёртка для вызова из веба.
Предварительно при запуске приложения прогружаются необходимые модули.
- 2. DGS: возникла трудность с окружениями.
Решение – установить переменную окружения.
Для веба вызывается отдельная функция, чем если бы мы вызывали напрямую из файла.

Интерфейс



Адаптивный дизайн



Что в итоге?

- Подход, основанный на правилах, остаётся эффективным для низкоресурсных языков.
- У подхода есть как преимущества, так и недостатки.
- Текущая реализация является лишь базой для перехода к более сложной архитектуре.

Дальнейшие планы

- Переход к нейросетевой архитектуре для генерации РЖЯ.
- Синтез видеоряда для РЖЯ по аналогии с немецким на основе поз.
- Расширение словаря глосс.
- Учёт гибкого порядка слов.



СПАСИБО ЗА ВНИМАНИЕ