Вот готовый файл README.md для проекта морфологического анализатора и корпуса азербайджанского языка.

**📘 README.md**

# 🇦🇿 Морфологический анализатор и корпус для азербайджанского языка

Этот проект представляет собой приложение на Python с графическим интерфейсом для морфологического анализа и аннотирования предложений. Он позволяет:

- Выполнять автоматический морфоанализ слов;

- Редактировать теги вручную;

- Добавлять аннотированные предложения в корпус;

- Экспортировать корпус в форматы CoNLL-U, CSV, JSONL и Excel.

---

## 📦 Структура проекта

morpho\_corpus/  
│  
├── annotator\_ui.py # UI-приложение (Gradio)  
├── text\_input.py # Токенизация и подготовка текста  
├── ml\_models.py # Предсказание морфологических тегов  
├── corpus\_db.py # Работа с файлом корпуса  
│  
├── export\_corpus.py # Экспорт в CoNLL-U  
├── export\_csv.py # Экспорт в CSV  
├── export\_jsonl.py # Экспорт в JSONL  
├── export\_excel.py # Экспорт в Excel (.xlsx)  
│  
└── corpus/  
└── corpus.json # Основной файл с корпусом

---

## 🚀 Установка

1. Установите зависимости (Python ≥ 3.8):

```bash

pip install gradio pandas openpyxl

1. Запустите интерфейс аннотации:

python annotator\_ui.py

Откроется окно в браузере: введите предложение, получите анализ, отредактируйте теги, добавьте в корпус.

**🧠 Обучение модели**

Модуль ml\_models.py содержит простую модель предсказания тегов. Вы можете заменить его на свой tagger, например с использованием BERT, CRF, или Neo4j-графов.

**📤 Экспорт корпуса**

После аннотирования экспортируйте корпус:

* CoNLL-U:

python export\_corpus.py

* CSV:

python export\_csv.py

* JSONL:

python export\_jsonl.py

* Excel (.xlsx):

python export\_excel.py

Все файлы будут сохранены в папке corpus/.

**🤝 Сотрудничество**

Проект открыт для совместной работы с лингвистами, особенно из стран тюркского мира. Цель — создать полнофункциональный NLP-инструментарий для агглютинативных языков.

**📜 Лицензия**

MIT License (для образовательных и исследовательских целей)

Вот обновлённый README.md с дополнительными секциями:

**📘 README.md (расширенный)**

# 🇦🇿 Морфологический анализатор и корпус для азербайджанского языка

Этот проект представляет собой приложение на Python с графическим интерфейсом для морфологического анализа и аннотирования предложений. Он позволяет:

- Выполнять автоматический морфоанализ слов;

- Редактировать теги вручную;

- Добавлять аннотированные предложения в корпус;

- Экспортировать корпус в форматы CoNLL-U, CSV, JSONL и Excel.

---

## 📦 Структура проекта

morpho\_corpus/  
├── annotator\_ui.py # UI-приложение (Gradio)  
├── text\_input.py # Токенизация и подготовка текста  
├── ml\_models.py # Предсказание морфологических тегов  
├── corpus\_db.py # Работа с файлом корпуса  
│  
├── export\_corpus.py # Экспорт в CoNLL-U  
├── export\_csv.py # Экспорт в CSV  
├── export\_jsonl.py # Экспорт в JSONL  
├── export\_excel.py # Экспорт в Excel (.xlsx)  
│  
└── corpus/  
└── corpus.json # Основной файл с корпусом

---

## 🚀 Установка

```bash

pip install gradio pandas openpyxl

**🖥 Запуск приложения**

python annotator\_ui.py

Откроется окно в браузере: введите предложение, получите анализ, отредактируйте теги, добавьте в корпус.

**🧠 Обучение и настройка модели**

Файл ml\_models.py содержит базовый шаблон для морфологического предсказания (например, через lookup-таблицу или ML-модель). Для продвинутой версии:

1. Соберите размеченный корпус (через annotator\_ui.py);
2. Экспортируйте в corpus/corpus.conllu;
3. Используйте данные для обучения BERT или CRF-модели:
4. # Примерно:
5. python train\_model.py --input corpus/corpus.conllu --output models/tagger.bin
6. Подключите свою модель в ml\_models.py.

**🧩 Расширение графа (NeoFUGI или Neo4j)**

Для поддержки агглютинативных языков морфоанализ основан на направленном графе:

* Узлы: корни, аффиксы, категории;
* Рёбра: допустимые трансформации;
* Атрибуты: грамматические признаки, вероятности.

Примерный стек:

* Схема: Neo4j или NetworkX
* Обучение: частотные правила, контекстные шаблоны
* Генерация: графовый поиск путей (BFS/DFS)

Файл ml\_models.py можно переписать на основе графовой модели.

**📤 Экспорт корпуса**

python export\_corpus.py # CoNLL-U

python export\_csv.py # CSV

python export\_jsonl.py # JSONL

python export\_excel.py # Excel (.xlsx)

Файлы будут сохранены в папке corpus/.

**🌍 Публикация корпуса**

1. Убедитесь в корректности: экспортируйте в CoNLL-U и проверьте формат.
2. Добавьте метаданные: язык, версия, лицензия, контакт.
3. Платформы для публикации:
   * [GitHub](https://github.com/)
   * [Zenodo.org](https://zenodo.org/)
   * [UniversalDependencies.org](https://universaldependencies.org/)
4. Лицензия: предпочтительно CC BY-SA 4.0 или MIT (если только инструмент).

**🤝 Сотрудничество**

Открыт для сотрудничества с:

* Лингвистами и исследователями тюркских языков;
* Разработчиками NLP-инструментов;
* Создателями цифровых образовательных платформ.

Связь: [указать email или ссылку на Telegram/Discord]

**📜 Лицензия**

MIT License — свободное использование для научных и образовательных целей.