**Тестовое задание 1**

**Популярные модели для распознавания текста**

1. **Google Web Speech API** (используется в recognize\_google):

* **Точность**: Высокая, благодаря обширным тренировочным данным и современным методам машинного обучения.
* **Скорость**: Очень высокая, так как распознавание выполняется на серверах Google.
* **Преимущества**: Поддержка множества языков, хорошая интеграция, высокая точность.
* **Недостатки**: Требуется подключение к интернету, потенциальные проблемы с конфиденциальностью данных.

1. **Microsoft Azure Speech-to-Text**:

* **Точность**: Высокая, благодаря использованию передовых технологий обработки речи.
* **Скорость**: Высокая, также осуществляется на удаленных серверах.
* **Преимущества**: Поддержка различных языков и акцентов, интеграция с другими сервисами Azure.
* **Недостатки**: Платная услуга, требуются настройки.

1. **IBM Watson Speech to Text**:

* **Точность**: Высокая, особенно для англоязычных текстов.
* **Скорость**: Высокая, выполнение на облачных серверах IBM.
* **Преимущества**: Мощные аналитические возможности, поддержка различных форматов.
* **Недостатки**: Платная услуга, возможно сложная интеграция.

1. **Vosk** (офлайн-модель):

* **Точность**: Средняя/высокая, зависит от модели и языка.
* **Скорость**: Высокая, выполняется на локальном устройстве.
* **Преимущества**: Не требует интернет-соединения, поддержка различных языков.
* **Недостатки**: Ограниченная точность по сравнению с онлайн-сервисами, необходимо загружать модели.

1. **CMU Sphinx** (офлайн-модель):

* **Точность**: Средняя, сильно зависит от языка и качества аудио.
* **Скорость**: Высокая, выполнение на локальном устройстве.
* **Преимущества**: Полностью офлайн, бесплатный и открытый исходный код.
* **Недостатки**: Ограниченная точность, особенно для сложных и шумных аудио.

### Сравнение моделей

| **Модель** | **Точность** | **Скорость** | **Поддержка языков** | **Требуется интернет** | **Преимущества** | **Недостатки** |
| --- | --- | --- | --- | --- | --- | --- |
| Google Web Speech API | Высокая | Очень высокая | Широкая | Да | Высокая точность, много языков | Конфиденциальность, требуется интернет |
| Microsoft Azure | Высокая | Высокая | Широкая | Да | Поддержка акцентов, интеграция | Платная услуга |
| IBM Watson | Высокая | Высокая | Широкая | Да | Мощные аналитические возможности | Платная услуга, сложная интеграция |
| Vosk | Средняя | Высокая | Средняя | Нет | Не требует интернет | Ограниченная точность |
| CMU Sphinx | Средняя | Высокая | Ограниченная | Нет | Бесплатный, открытый исходный код | Ограниченная точность |

**Выбор лучшей модели**

С учетом требований высокой точности и скорости распознавания текста, а также необходимости в простоте интеграции и поддержке множества языков, **Google Web Speech API** является лучшим выбором для большинства приложений. Это объясняется следующими факторами:

* **Высокая точность**: Google использует одни из самых передовых технологий машинного обучения и имеет доступ к огромному количеству данных для тренировки моделей.
* **Скорость**: Распознавание происходит на мощных серверах, обеспечивая быстрый отклик.
* **Поддержка множества языков**: Поддержка широкого спектра языков и акцентов.
* **Интеграция**: Простота использования через библиотеку SpeechRecognition и функцию recognize\_google().

**Тестовое задание 2**

**Выбранные методы и модели**

**1. Запись и сохранение аудио**

Для записи и сохранения аудио использовались:

* **JavaScript для записи аудио с микрофона:** Этот скрипт интегрирован в HTML и позволяет пользователю записывать аудио напрямую в браузере.
* **FFmpeg для конвертации аудио:** Использование FFmpeg для конвертации записанного аудио в формат WAV, который подходит для дальнейшей обработки.

**2. Сегментация аудио по говорящим (диаризация)**

Для сегментации аудио на части, соответствующие разным говорящим, использовалась модель из библиотеки **Pyannote**:

* **Pyannote.audio:** Это библиотека для обработки аудио и выполнения задач, таких как диаризация, распознавание речи и другие. Она предоставляет предобученные модели для сегментации аудио.
* **Pipeline 'pyannote/speaker-diarization':** Используется для выполнения диаризации, которая разбивает аудиофайл на части, каждая из которых соответствует определенному говорящему. Модель тренируется на большом количестве данных и может эффективно разделять аудио по говорящим.

**3. Распознавание речи**

Для распознавания речи в каждом сегменте использовалась библиотека **SpeechRecognition**

**Обоснование выбора**

**1. Диаризация с использованием Pyannote.audio**

* **Точность:** Pyannote.audio предоставляет одну из лучших моделей для задачи диаризации с высокой точностью, что крайне важно для правильного разделения аудио по говорящим.
* **Легкость использования:** Предобученные модели и простота интерфейса позволяют легко интегрировать диаризацию в рабочий процесс.
* **Поддержка аутентификации:** Возможность использования токенов аутентификации для доступа к моделям на платформе Hugging Face, что обеспечивает безопасность и контроль доступа.

**2. Распознавание речи с использованием SpeechRecognition**

* **Доступность:** SpeechRecognition и Google Web Speech API предоставляют бесплатный доступ с хорошей точностью распознавания речи для множества языков, включая русский.
* **Простота интеграции:** Библиотека имеет простой и понятный интерфейс, который легко интегрировать с другими компонентами системы.
* **Широкое использование:** Google Web Speech API широко используется в различных приложениях и обладает высокой точностью распознавания речи благодаря большим объемам данных, на которых он тренируется.

**Заключение**

Выбор методов и моделей для данной задачи был сделан на основе их точности, простоты использования и доступности. Pyannote.audio предоставляет эффективные инструменты для диаризации, а SpeechRecognition с Google Web Speech API обеспечивает высокую точность распознавания речи. Комбинированное использование этих библиотек позволяет достичь цели задачи — сегментации аудио по говорящим и распознавания речи в каждом сегменте — с высокой точностью и эффективностью.