# Документация по решению

**Название кейса:** Контроль и управление изменениями в тендерных закупках.

### **Тизер решения**

Мы разработали эффективное решение для анализа и сопоставления требований в тендерных закупках с использованием локальных моделей LLM. Решение включает продвинутые подходы к обработке текста с помощью промптинга, что позволяет точно выявлять соответствие между требованиями и спецификациями. Модели работают как на **CPU**, так и на **GPU**, что значительно ускоряет обработку данных (время обработки одной пары документов сокращается с 1 минут при CPU до 4,5 секунд при использовании GPU).

Для локальной оценки решения и настройки модели произведена аугментации данных, что позволило улучшить результаты.

В результате, наш подход позволяет эффективно анализировать документы, обеспечивая высокую точность и стабильность при различных запусках.

### **ОПИСАНИЕ РЕШЕНИЯ**

### **Дополнительные материалы:**

* **Анализ экспериментов:**[Ссылка на таблицу анализа экспериментов](https://docs.google.com/spreadsheets/d/1nQyicVKDL6OcQBWqS-yTfViVwztdxVKllaEJv_1k5p0/edit?usp=sharing)
* **Определены общие блоки (подразделы) по документам:**[Ссылка на таблицу с блоками](https://docs.google.com/spreadsheets/d/153ZVY9afFlBTPNctP-CW9XFLEWjo-vVXc1nIRJLZmJI/edit?gid=0#gid=0)

### 

### **Baseline**

Создан общий *prompt* для локальной LLM, в который подаются два документа целиком и требуется провести сопоставление и анализ текста на предмет соответствия требований (UC) и спецификации (SSTS). Ожидаемый результат от LLM — данные по всем полям:

* **Name**
* **Differences**
* **Description**
* **Compliance Level**

Бейзлайн дал результат **0.22222** (при **MSE 1.666**) на тренировочных данных.

### **Анализ по блокам**

Рассматривались различные варианты деления документа:

* По предложениям
* По абзацам/параграфам
* По разделам
* Предварительное деление с помощью LLM

В результате было принято решение использовать деление по разделам как оптимальное, так как оно быстрее, чем LLM, и более эффективно, чем другие методы, кроме LLM.

### **Разбиение документа на блоки (подразделы)**

Определены общие блоки (подразделы) для документов. Пример для UC (Use Case):

* **Use Case**
* **Description**
* **Goal**
* **Context**
* **Scope**
* **Actors**
* **Triggers**
* **Preconditions**
* **Main Scenario**
* **Postconditions**
* **Requirements**
* **Alternative Scenario**
* **Components**
* **Function logic**

Стоит отметить, что в текстах часто встречаются опечатки, даже в наименованиях подразделов. Также разделы могут быть расположены не по порядку. С разбиением на блоки можно ознакомиться по [ссылке](https://docs.google.com/spreadsheets/d/153ZVY9afFlBTPNctP-CW9XFLEWjo-vVXc1nIRJLZmJI/edit?gid=0#gid=0).

### **Алгоритм автоматического определения блоков**

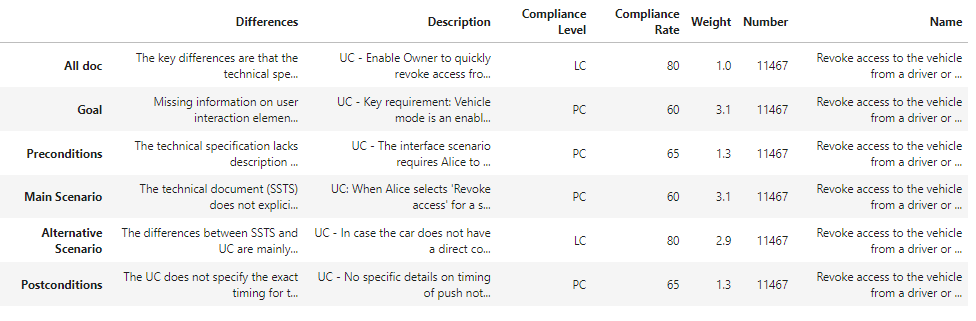
Для исключения проблем с опечатками автоматическое определение блоков (разделов) осуществляется по следующему алгоритму:

1. Деление всего текста на абзацы (параграфы) средствами разбора документов типа *.docx* (так как текст изначально представлен в виде абзацев).
2. Перевод каждого абзаца в эмбеддинг с помощью произвольной модели-эмбеддера (основанной на кодере BPE).
3. Сравнение косинусной близости абзаца с предопределенным названием раздела. Если косинусное сходство более **0.9**, то считается, что найден матчинг.
4. На основе идентификаторов абзацев формируется словарь "наименование раздела" - "текст раздела".
5. Далее этот словарь используется для дальнейшего определения соответствия требованиям.

### **Процесс анализа соответствия разделов**

При анализе сопоставления разделов используется *prompt*, который, помимо полей **Name**, **Differences**, **Description**, **Compliance Level**, также определяет **Compliance Rate** (численное значение сопоставления от 0 до 100). Также производится сравнение всего документа, и этому расчету присваивается вес. Все оценки сравнения переводятся в единую таблицу результатов по каждому документу.

**Пример результата (черновик):**



### **Проблемы с нестабильностью результатов**

Наблюдались проблемы, что даже при низкой температуре генерации LLM результаты при разных запусках могли сильно меняться. Для стабилизации результатов были задействованы дополнительные параметры, такие как **top\_k**, **top\_p** и другие.

### **Формирование сводной оценки по документу**

**Итоговый Complience Rate** (значение от 0 до 100) для документа определяется как взвешенное усреднение, т.е. процент соответствия усредняется в зависимости от значения **веса (Weight)**.

### **Итоговый Differences – два варианта расчета:**

* **High**: При наличии ресурсов и времени на обработку производится объединение всех выявленных **Differences**, после чего производится суммаризация до сводного значения **Differences**.
* **Low**: Используется тот **Differences**, который определен для общего документа (“All doc”).

### **Итоговый Description – два варианта расчета:**

* **High**: При наличии ресурсов и времени на обработку производится объединение всех выявленных **Description** (включая итоговый **Differences** и исходные тексты), после чего проводится суммаризация итогового значения **Description**. Идея заключается в том, чтобы изначально выявить **Description** для каждой части текста, рассчитать итоговый **Differences** (см. выше), а затем применить общий *prompt* со всеми данными для определения итогового **Description**. Этот метод требует больше времени и токенов.
* **Mid**: При наличии ресурсов и времени на обработку производится объединение всех выявленных **Description** и далее суммаризация до сводного значения **Description**.
* **Low**: Используется тот **Description**, который определен для общего документа (“All doc”).

### **Аугментация данных**

Данных для обучения очень мало, поэтому рассматривались несколько вариантов аугментации:

1. **Найти аналогичные документы в сети Интернет**:  
   Есть возможность найти реальные данные, однако есть сомнения относительно их формата. Первичный поиск не дал результатов.
2. **Сгенерировать через ChatGPT/Claude**:  
   Вопрос заключается в том, что сгенерированные данные могут не отражать действительность, и возможно наличие смещения в данных.
3. **Сгенерировать данные на основе существующих**:  
   Это оптимальный выбор, хотя возможна проблема, что структура в файлах **SSTS** может отличаться от реальных заявок и быть просто копией файлов **UC**.

**Метод генерации положительных и отрицательных примеров:**

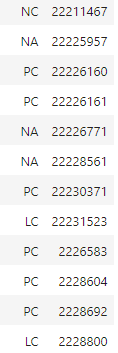
* **Положительные примеры** (полное совпадение):  
  Берутся исходные файлы **UC** и просто создаются их копии, что гарантирует покрытие всех требований.
* **Негативные примеры**:  
  Берутся файлы **SSTS** и перемешиваются с идеей, что различные спецификации могут быть подложены к разным требованиям **UC**.

Этот подход позволил увеличить датасет в 3 раза:

* Исходные данные — 12 шт.
* Положительные примеры — 12 шт.
* Негативные примеры — 12 шт.

### **Результаты аугментации данных**

Увеличение данных позволило более точно оценивать модели и решения в целом. Результаты моделей стали более стабильными. Дополнительно введение аргументированных данных позволило провести оценку корректности бейзлайн модели для понимания возможных улучшений.

Например, **бейзлайн модель** для положительных примеров (начинаются с “111”) чаще указывала на **FC/LC/PC**, а для негативных примеров (начинаются с “222”) чаще указывала на **PC/NC/NA**.

### **Скорость расчета**

Переход от **CPU** на **GPU** дал значительное ускорение — расчет стал занимать **54 секунды** вместо **12 минут**. Однако не у всех членов команды есть видеокарты, поэтому разработка логики решения велась параллельно для обеих платформ: **CPU** и **GPU**.

### **Влияние гиперпараметров**

По итогам анализа проведенных экспериментов ([ссылка на таблицу](https://docs.google.com/spreadsheets/d/1nQyicVKDL6OcQBWqS-yTfViVwztdxVKllaEJv_1k5p0/edit?gid=0#gid=0)) было установлено, что настройки гиперпараметров при вызове локальных LLM оказывают большое влияние на результаты. К таким параметрам относятся:

* **Температура**
* **top\_k / top\_p** — параметры, управляющие правилами генерации следующего слова (например, **beam search**)
* **Штрафы** — такие как **repeat\_penalty** и другие

### **Килл-фичи**

**Деление на разделы**Для оценки важности каждого раздела были использованы различные методы. Рассматривались следующие подходы для оценки значимости разделов:

* **Сходство общего эмбеддинга**
* **Среднее сходство по предложениям**
* **Количество похожих объявлений**
* **Косинусное сходство**

Анализ применения методов семантического сходства

| Number | Compliance Level | Most sentence similarity | BERT similarity | Sentence-BERT similarity | Average sentence similarity |
| --- | --- | --- | --- | --- | --- |
| 11467 | FC | 3 | 0,908 | 0,903 | 0,763 |
| 8800 | LC | 9 | 0,895 | 0,923 | 0,834 |
| 8692 | LC | 12 | 0,920 | 0,907 | 0,817 |
| 26161 | LC | 1 | 0,833 | 0,794 | 0,802 |
| 26771 | LC | 11 | 0,800 | 0,894 | 0,801 |
| 28561 | LC | 8 | 0,891 | 0,901 | 0,840 |
| 30371 | LC | 3 | 0,929 | 0,927 | 0,874 |
| 31523 | LC | 2 | 0,739 | 0,830 | 0,833 |
| 25957 | PC | 1 | 0,738 | 0,809 | 0,843 |
| 6583 | NC | 15 | 0,958 | 0,950 | 0,838 |
| 8604 | NC | 6 | 0,911 | 0,876 | 0,818 |

Однако эти методы не позволяют четко сопоставить требования. Например, очень похожие документы на самом деле могут иметь значение **NC** (не соответствует), что указывает на то, что недостаточно просто искать текстовые сходства. Необходимо учитывать именно сходство требований, а не только общие текстовые характеристики.

Также был проверен метод сходства через **TF-IDF**, однако он не учитывает контекст документов. Для улучшения результатов требуется более глубокий анализ с учетом контекста. А также для создания качественного словаря требуется увеличить количество документов

### **Итоговое решение**

На основе **промптинга** и **LLM** (локальных моделей) генерация требуемого результата осуществляется в заданном формате, который затем распарсивается в итоговую таблицу.

**Итоговое решение** позволяет выполнять код как с использованием **CPU**, так и с использованием **GPU**. Использование **GPU** значительно сокращает время обработки, обеспечивая ускорение расчетов.

### **Что улучшить:**

* Реализовать подходы уровня **“High”**, которые были описаны выше.
* Увеличить объем данных для более точной оценки и обучения модели.
* Создание идеальных промтов под текущею задачу для открытых LLM
* Подготовка расширенного датасета и обучение Lora адаптера
* Создание инструкций для классификаций похожести текстов с привлечением максимального количества экспертов
* Полный фантюнинг своей LLM под поставленную задачу