**Тестовое задание**

1. **Детекция ошибок**. Один из этапов обработки данных – разметка документов для обучения модели. Этим занимаются специально обученные разметчики, которые, ввиду человеческого фактора, могут совершать различного рода ошибки (опечатываться, присваивать сущностям неправильные теги и т.п.).   
     
   Для выполнения задания предлагаем ознакомиться с несколькими документами типа «Протокол» в формате html, которые были получены в результате распознавания исходных отсканированных документов. К документам прилагается таблица с информацией об их разметке.   
     
   В колонке «DocName» таблицы содержатся названия документов. В колонке «Tokens» — разбитые на токены тексты исходных документов. Для каждого токена представлена разметка по двум целям:

* Выделение именованных сущностей, классический NER (в имени колонки префикс «КЕ»);
* Сегментация документа на различные семантические блоки (в имени колонки префикс «Blocks»).

В таблице представлена многоуровневая разметка именованных сущностей и блоков: то есть в рамках одной сущности/блока могут содержаться другие сущности/подблоки. Например, сущность Адрес может состоять из СубъектРФ, Улица, Дом. Кроме того, именованные сущности могут пересекаться. Префикс «B-» обозначает начало, а «I-» — середину или конец сущности. При этом не все сущности имеют префиксы и, если в какой-либо из колонок их нет, это означает, что нам не нужно различать одинаковые сущности, следующие друг за другом.

Таким образом, каждая колонка таблицы соответствует разным типам и уровням сущностей/блоков; сущность может состоять из произвольного числа токенов; а каждый токен может иметь множество меток.

После ознакомления с данной выборкой:

1) Предложите, какую аналитику и визуализацию можно сделать по этим данным? Приведите примеры реализации в коде;

2) Для одной сущности на выбор из каждой цели разметки (Blocks и KE) реализуйте подсчет статистик, с помощью которых можно выявить возможные ошибки в данных.

1. **Prompt-engineering.** Генеративные модели активно используются в задачах LegalTech. Предлагаем вам решить задачу на prompt-engineering. Необходимо извлечь (при наличии) следующие атрибуты из документа dkp.txt, используя любую LLM, библиотеку LangChain и навыки промпт-переговорщика:

* **СтороныДоговора[]** – необходимо извлечь информацию о сторонах договора:
  + ТипСтороны – выбор из значений «Продавец» и «Покупатель»
  + РосЮЛ – наименование юридического лица, если оно выступает стороной сделки. Подача в им. падеже.
  + ФИО – наименование физического лица, если оно выступает стороной сделки. Подача в им. падеже.
  + ИНН – инн стороны сделки
  + ОГРН – огрн стороны сделки
  + Адрес – адрес регистрации стороны сделки
    - Страна
    - Субъект
    - НаселенныйПункт
    - Улица
    - Дом
* **ПредметЮЛ** – информацию об юридическом лице, долю которого передают по договору.
  + Наименование. Подача в им. падеже.
  + ИНН
  + ОГРН
  + КПП
  + Адрес
    - Страна
    - Субъект
    - НаселенныйПункт
    - Улица
    - Дом
* **ДоговорнаяСтоимость** – это реальная стоимость отчуждаемой доли, за которую стороны договорились произвести отчуждение. Необходимо, чтобы ответ был преобразован в следующий формат: X…X,YY. Где X - рубль, Y – копейка.
* **НоминальнаяСтоимость** – это номинальная стоимость отчуждаемой доли в уставном капитале. Необходимо, чтобы ответ был преобразован следующий формат: X…X,YY. Где X - рубль, Y – копейка.
* **ПоказательДоли** – это размер отчуждаемой доли / части доли в уставном капитале. Необходимо, чтобы ответ был преобразован в следующий формат: указывается значение без знаков и пробелов, дробная часть отделяется запятой, округление производится до 6 знаков после запятой.

**(!)** Важно получать предсказуемый ответ, чтобы можно было его обрабатывать дальше в коде (например, без «*я думаю, что договорная стоимость составляет* 1 руб»). Подумайте над выходным форматом ответа модели. Также важно в запросах проверять наличие в документе каждого атрибута.

**(!!)** Каждая модель имеет ограниченное окно внимания, подумайте над способом фрагментации текста, чтобы подавать в модель контролируемое количество токенов.

Ждём результатов выполнения задания в виде Jupyter ноутбука с развёрнутыми комментариями. Удачи!