команда
против пыток

# Язык пыток

## Приложение 2
Технический алгоритм того, как мы анализировали тексты на количественном этапе исследования

18+

Настоящий материал (информация)
произведен, распространен и (или)
направлен иностранным агентом
«Команда против пыток»
либо касается деятельности
иностранного агента
«Команда против пыток»

18+

На этапе предобработки текстов судебных решений и материалов СМИ мы проделали следующие операции:

1) для унификации написания привели регистр всех текстов к нижнему;
2) частично удалили из текстов пунктуацию (оставили лишь знаки препинания, обозначающие конец предложений);
3) удалили арабские цифры;
4) лемматизировали тексты;
5) удалили стоп-слова (русские и английские слова, усложняющие анализ и не дающие веса при оценке содержания написанного, например, названия документов, служебные слова и т. д.); для анализа материалов средств массовой информации мы при помощи стоп-слов отсекли тексты, связанные с пытками в бытовом или военном контексте;
6) удалили типичные для судебной системы стоп-слова (часто встречающиеся в текстах слова и формализмы, кочующие из решения в решение; частично список таких слов позаимствован у «Алгоритма света»);
7) удалили двойные пробелы.

## 1. Анализ частотности

После первичной предобработки текста мы проанализировали частотность использования определённых терминов. Для того чтобы найти важные термины, применялась TF-IDF векторизация. В отличие от простого подсчёта слов такая векторизация:

- считает, как часто термин появляется относительно общего количества терминов в документе (TF — term frequency). Например, если слово «военный» появляется в документе три раза на каждые 100 слов, значит TF = 0,03;
- оценивает, насколько редок определённый термин во всём корпусе анализируемых текстов (логарифм от отношения общего количества документов к количеству документов, содержащих термин +1, IDF — inverse document frequency). Допустим, слово «военный» встречается в десяти текстах из 1000. Тогда IDF = ln(1000/10+1) ~ 4,5. Чем реже встречается слово, тем выше IDF и ниже TF.

Далее мы умножаем эти две метрики и получаем показатель TF-IDF. Чем выше оценка, тем более часто термин встречается в конкретном документе, но при этом не распространён во всём корпусе текстов. По этой оценке мы и сравнивали упоминания слов в каждом корпусе в целом, а также проводили анализ словоупотреблений внутри каждого календарного года.

Такой подход решает сразу две задачи: (1) мы опускаем в рейтинге распространённые слова (союзы, предлоги, типичные канцеляриты и т. д.) и (2) подсвечиваем важные слова, которые встречаются не так часто, но имеют значение для анализа. Также TF-IDF векторизация позволяет анализировать контекст и оценивать тематику корпуса, не читая каждый текст по отдельности.

Кроме того, мы проводили векторизацию итеративно, каждый раз находя бессмысленные термины (например: мм, дд, ук, рф), которые пополняли наш список стоп-слов.

Далее мы провели анализ текстов на предмет последовательного неразрывного сочетания тех или иных слов. Попросту говоря, мы проверили решения через модель векторизированных n-грамм, где n — число последовательно соединяющихся слов. Проблема анализа обособленных слов и вычисления их частотности заключается в том, что отдельному термину может не доставать контекста, поэтому его интерпретация может быть двоякой. N-граммы такой проблемы не имеют. Например, слово «военный» может трактоваться как существительное или прилагательное, но если мы анализируем биграммы (сочетания двух слов, от лат. би — «двойной»), то мы увидим, что термин «военный» часто находится рядом с термином служба.

**Пример алгоритма поиска n-грамм**

- Мы имеем исходное предложение: «Военный ударил рукой по лицу»;
- приведённое к одному регистру и лемматизированное предложение (с приведением слов к словарной форме) после удаления служебных слов будет выглядеть так: «военный ударить рука лицо»;
- список получившихся униграмм (отдельных слов): военный, ударить, рука, лицо;
- биграммы (пары): военный ударить, ударить рука, рука лицо;
- триграммы (последовательности трёх слов): военный ударить рука, ударить рука лицо.

Таким образом предложение можно привести до анализируемых последовательностей практически до бесконечности, если это позволяют анализируемые тексты.

## 2. Сентимент-анализ

Метод сентимент-анализа предназначен для определения эмоциональной окраски текстов с использованием модели машинного обучения. Он включает разбиение длинных текстов на части, обработку их по модели сентимент-анализа и объединение результатов для получения итогового сентимента с учётом веса уверенности модели. Использование этого подхода позволяет понять, в каком свете предстаёт в тексте описываемое событие — в позитивном, негативном или нейтральном. Ниже приводится описание всех шагов по работе с этой моделью.

**Ниже приводится описание всех шагов по работе с этой моделью**

1) **Загрузка и настройка модели**: используется предобученная модель сентимент-анализа «blanchefort/rubert-base-cased-sentiment», разработанная для русского языка.
2) **Обработка текстов.** Сначала каждый текст разбивается на фрагменты (chunks) длиной не более 500 токенов с помощью токенизатора. Токен — это минимальная единица текста, с которой работает модель. Обычно это

отдельное слово/символ. Ограничение по подгружаемым токенам необходимо для корректной работы модели, которая имеет ограничение на длину текста. Далее для каждого фрагмента текст декодируется обратно из токенов в строку.

3) **Анализ сентимента.** Модель анализирует сентимент каждого фрагмента, возвращая метку (POSITIVE, NEGATIVE, NEUTRAL) и confidence score (CS). CS — это метрика модели, число, которое показывает, насколько модель уверена в своём предсказании. Оно измеряется от 0 до 1, где 0 — модель совсем не уверена, 1 — модель полностью уверена. В этом измерении 1 означает 100-процентную уверенность модели в своём предсказании.

Например, у нас есть текст: «Этот фильм просто потрясающий!»

Модель говорит: «Это позитивный текст, и я уверена в этом на 97%».

Эти 97% — это и есть confidence score, который будет равняться показателю 0,97 из 1. Чем ближе к 100%, тем больше можно доверять результату. Если результаты ниже 50%, то модель сомневается, а результат стоит перепроверить вручную.

CS определяется моделью автоматически на основании анализа текстов, которые она видела ранее. По каждому из вариантов (позитивный, нейтральный или отрицательный) модель выносит свой CS и выдаёт тот ответ их трёх, по которому он выше.

Итоговый сентимент текста определяется взвешенным средним значением сентимента всех фрагментов, разделённых на токены:

- для расчёта учитывается уверенность модели в каждом фрагменте (чем выше уверенность, тем больший вес имеет предсказание);
- результат взвешивания отображается в виде одного из трёх классов: POSITIVE, NEGATIVE или NEUTRAL, с указанием итогового уровня уверенности.

4) **Обработка и сохранение данных по пакетам.** Для оптимизации работы тексты обрабатываются частями (batch_size=512). В каждом батче тексты проходят анализ сентимента, результаты включают (1) оригинальный текст, (2) год написания текста и (3) итоговый сентимент с уровнем уверенности в предсказании. Итоги обработки сохраняются в формате CSV по пакетам.