

Московский государственный университет имени М. В. Ломоносова  
Факультет вычислительной математики и кибернетики  
Кафедра математических методов прогнозирования

Косарев Евгений Александрович

## НЕЙРОСЕТЕВЫЕ МОДЕЛИ ЯЗЫКА ДЛЯ ВЫЯВЛЕНИЯ РЕЧЕВЫХ МАНИПУЛЯЦИЙ В НОВОСТНЫХ ПОТОКАХ

Выпускная квалификационная работа

**Научный руководитель:**  
д.ф.-м.н., профессор  
*К.В.Воронцов*

Москва, 2022

# Содержание

<b>1</b>	<b>Введение</b>	<b>2</b>
<b>2</b>	<b>Постановка задачи</b>	<b>3</b>
<b>3</b>	<b>Разметка</b>	<b>4</b>
3.1	Сбор и обработка данных . . . . .	4
3.1.1	Как устроен NER . . . . .	5
3.2	Структура манипуляций . . . . .	6
3.3	Подробнее о разметке . . . . .	7
3.3.1	Методика разметки . . . . .	7
3.3.2	Шаблон разметки . . . . .	8
3.4	Анализ размеченной выборки . . . . .	9
<b>4</b>	<b>Двухуровневая модель</b>	<b>12</b>
4.1	Декомпозиция задачи . . . . .	12
4.2	Обучающее множество . . . . .	13
4.3	Оценка качества . . . . .	14
4.4	Описание экспериментов . . . . .	15
4.4.1	NER и RE . . . . .	16
4.4.2	NER_BERT, NER_BERT+LL . . . . .	16
4.4.3	RE, RE+LL . . . . .	17
4.5	Результаты . . . . .	17
<b>5</b>	<b>Модификация модели второго уровня</b>	<b>19</b>
5.1	Описание экспериментов . . . . .	20
5.1.1	Dual-encoder . . . . .	20
5.1.2	Cross-encoder . . . . .	21
5.1.3	In-batch техника . . . . .	21
5.1.4	Негативное сэмплирование и функция потерь . . . . .	22
5.2	Результаты . . . . .	23
<b>6</b>	<b>Выбор гиперпараметров</b>	<b>24</b>
<b>7</b>	<b>Заключение</b>	<b>25</b>

# 1 Введение

Русскоязычные СМИ ежедневно создают внушительный объем данных: генерируют поток новостей, достигающий сотен тысяч сообщений в сутки. Новости могут отличаться содержанием, жанром, приемами подачи информации. При этом каждый новостной ресурс ставит перед собой цель привлечь как можно более широкую аудиторию.

На сегодня задача поиска фейков [1], [2] (недостоверной информации) является весьма актуальной. В мире работает более 350 крупных агентств, профессионально занимающихся выявлением и анализом недостоверной информации. Их работа осложняется большими объемами информации, которая поступает к ним ежедневно. Недостоверная данные могут приходиться из социальных сетей, рекламных роликов, слухов, а также недобросовестных СМИ [3].

Существует несколько способов определить степень недостоверности новостного сообщения. Например, в [6] производится перекрестное сравнение информации, а именно поиск противоречий среди различных описаний одного и того же явления, и наиболее конфликтующее с остальными из них объявляется недостоверным. Возможен подход по оцениванию эмоциональной окрашенности текста [5]. При таком предположении текст, в котором автор придает повествованию эмоциональный окрас, с большей вероятностью содержит необъективную и, возможно, ложную информацию. Часто используются графы знаний, позволяющие структурировать данные и производить быстрый анализ поступающей новой информации [7].

Несмотря на многообразие приемов определения недостоверной информации, к сожалению, проверка текста на достоверность все еще проводится вручную, хоть и с привлечением различных алгоритмов. На текущий момент большая часть из них, например, поиск противоречий, основана на внешней информации. Однако подобные алгоритмы, в том числе графы знаний, быстро устаревают [8] ровно в тот момент, когда приходится обновлять информацию или добавлять новые данные. Выгодно смотрятся методы, не учитывающие или практически не учитывающие информацию, которой нет в тексте. Таким образом, результаты их работы зависят лишь от тональности слов и предложений, от языковых конструкций и не требуют постоянного перестроения базы знаний или её дообучения.

Один из таких способов оценить достоверность новостного текста — провести количественный анализ языкового манипулирования. По определению, **манипулирование** — это вид речевого воздействия, целью которого является неявное внесение в психику адресата чуждых ценностей, желаний и целей с использованием психолингвистических механизмов. С помощью такого приема можно достичь нейтрального, позитивного или негативного отношения к мишени, по отношению к которой ведется манипулирование. В этой работе исследуется возможность применения алгоритмов машинного обучения для поиска речевых манипуляций.

## 2 Постановка задачи

Будем пользоваться следующей терминологией:

- **Новостной текст, новость** — отдельный текст из новостного потока, опубликованный на русском языке.
- **Речевая/языковая манипуляция** — фрагмент новостного текста, который содержит в себе речевое воздействие на объект манипуляции.
- **Объект манипуляции, мишень манипуляции, сущность** — объект в тексте, на который направлено влияние манипуляции, целью которого ставится изменить отношение читателя к данному объекту.

Таким образом, целью данной работы является создание алгоритма, который получает на вход новостной текст на русском языке и выделяет в нем следующие тройки:

1. Подстрока, фрагмент текста, содержащий языковую манипуляцию
2. Тип манипуляции, который содержится в этой подстроке
3. Подстрока, фрагмент текста, который является мишенью манипуляции

Пример выделенной тройки:

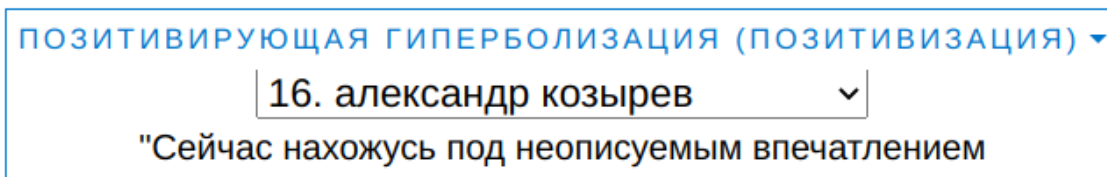


Рис. 1: Пример разметки в Яндекс Толоке

Здесь "позитивирующая гиперболизация (позитивизация)" — тип манипуляции, "александр козырев" — мишень, на которую направлена манипуляция, а остальное — фрагмент текста, содержащий манипуляцию.

Предлагается основой алгоритма выбрать заранее предобученные на русскоязычных текстах нейросетевые модели, например, из библиотеки DeepPavlov<sup>1</sup> и обучать их определять необходимые тройки в новостных текстах. Для достижения необходимых результатов будут решено несколько задач: от создания нужной разметки новостных текстов до построения модели машинного обучения, которая будет производить поиск и классификацию манипуляций.

В задаче разметки будет проанализирована структура новостных сообщений, собранных с таких ресурсов, как *currenttime*, *rbc*, *ria news*, *tass*, *meduza* и *другие*. Будет

<sup>1</sup><http://docs.deeppavlov.ai/en/master/index.html>

выделена группа наиболее популярных способов манипуляций и составлена методика их поиска и разметки. Также будет проведен качественный анализ полученной разметки.

На вход алгоритмам машинного обучения подается новостной текст и заранее выделенные именованные сущности (потенциальные объекты манипуляции). Будет осуществлено несколько подходов по построению модели поиска и классификации манипуляций, а еще составлена методика по оценке качества работы этих алгоритмов.

## 3 Разметка

Для обучения алгоритма машинного обучения, а также финальной оценки качества модели необходимо собрать и подготовить данные, по которым можно будет провести построение итоговой модели. Алгоритм создания разметки можно разделить на несколько последовательных этапов:

- Сбор и первичная обработка данных.
- Анализ собранных новостных текстов и составление формальной структуры манипуляций, встречающихся в новостных текстах.
- Подготовка шаблона разметки в Яндекс Толоке и разметка данных.
- Анализ получившейся разметки.

### 3.1 Сбор и обработка данных

Был проведен анализ в результате которого выявлен набор СМИ от самых рейтинговых и популярных до желтых СМИ. Были разработаны программы, позволившие собрать из новостного потока тексты новостей за период с осени 2021 по зиму 2022 года с таких ресурсов, как *currenttime*, *rbc*, *ria news*, *tass*, *meduza* и другие, всего 19 источников. Из-за особенностей работы алгоритмов и html кода web-страниц сайтов, тексты требовали предобработки. Поэтому они проверялись на нарушения кодировки, наличие служебных тэгов и прочих технических артефактов в процессе скачивания их с интернет-ресурсов.

Из полученного текста с помощью предобученного алгоритма NER (Named Entity Recognition) [9] из библиотеки DeepPavlov производится выделение именованных сущностей в тексте. Среди этих именованных сущностей будут искажаться мишени манипуляции. Так же было учтено, что манипуляции внутри текста могут быть направлены на одну и ту же мишень. Однако, одна и та же манипуляция не может быть направлена на две различные мишени. На следующий этап подаются тексты со списком выделенных в них именованных сущностей.

### 3.1.1 Как устроен NER

Во всех будущих моделях входной текст разбивается на токены, каждый токен можно ассоциировать с частью слова или с самим словом. Назовем  $x$  — входной последовательностью токенов. Тогда:

$$NER : x \rightarrow y,$$

где  $y$  — последовательность меток, равная по длине  $x$

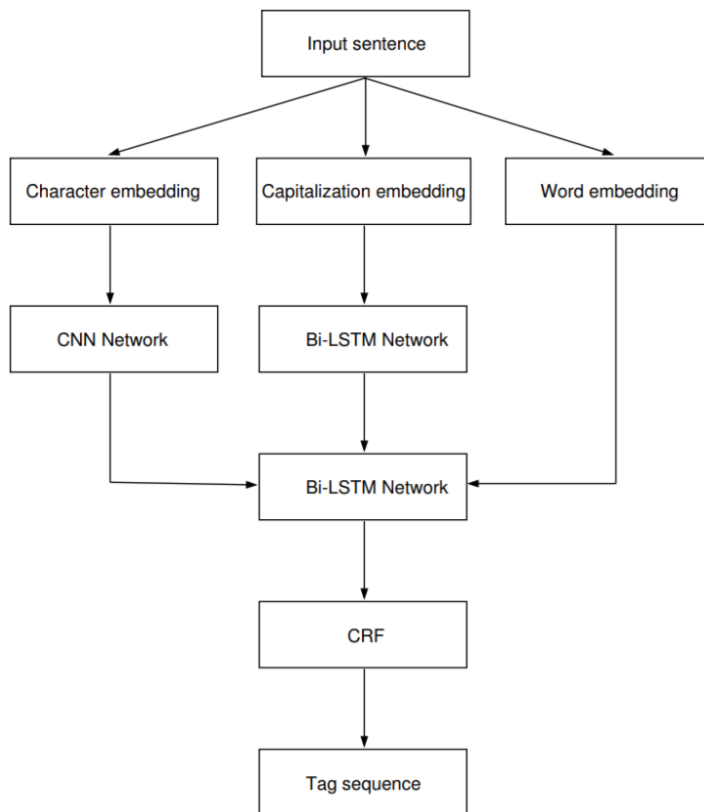


Рис. 2: Схематическое устройство применяемой модели NER

Последовательность токенов (Input sentence) подается на вход трем алгоритмам, каждый из них подсчитывает определенные признаки, которые конкатенируются в итоговый вектор признаков:

1. **Character embedding** — не каждый токен находится в исходном словаре, поэтому используется сверточная нейронная сеть (CNN). Она фиксирует морфологическую информацию об исходном представлении токена, такую как префикс и суффикс [10, 11]. Итоговая информация агрегируется в числовой вектор.
2. **Capitalization embedding** — каждый токен сопровождается меткой, отражающей информацию о наличии заглавных букв в исходном строковом представлении токена. С помощью модели Bi-LSTM [12] эта информация агрегируется в числовой вектор.

3. **Word embedding** — модель-векторизатор токенов (например, fastText [14]), которые есть в словаре.
4. Полученные числовые векторы конкатенируются и обрабатываются последовательно, модель CRF [13] выдает распределение, после которого каждому токену назначается наиболее вероятная метка.

## 3.2 Структура манипуляций

Проанализировав более тысячи новостных текстов вместе с группой лингвистов из РУДН, получилось выделить наиболее часто встречающиеся в новостных текстах манипуляции. По итогам этого исследования манипуляции чаще всего можно найти в:

- Новостных текстах, содержающие политические сводки;
- Новостных текстах об противоправных инцидентах;
- Новостных текстах, освещающих общественно резонансные события.

Опуская лингвистические особенности исследования, хочется сосредоточиться на той уникальной структуре типов манипуляции, которую удалось составить. Было решено представить иерархическую двухуровневую структуру. На верхнем уровне располагаются четыре группы речевых манипуляций, каждая из которых характеризует наиболее общие группы манипуляций. В свою очередь на втором уровне содержатся уточняющие типы манипуляций к четырём группам, они позволяют приобрести более глубокое понимание цели и способа манипуляции, уловить более тонкие оттенки смысла. Предложенная структура представлена ниже:

### 1. Негативизация

- 1.1 Навешивания ярлыков
- 1.2 Дисфемизмы
- 1.3 Поставка мишени в один ряд с негативно оцениваемым объектом
- 1.4 Антифразис
- 1.5 Прием обесценивания
- 1.6 Негативирующая гиперболизация
- 1.7 Прием моделирования негативного сценария
- 1.8 Вкрапление депрессивов

### 2. Позитивизация

- 2.1 Эвфемизация
- 2.2 Лозунговые слова и словосочетания

## 2.3 Позитивирующая гиперболизация

### 3. Деавторизация

3.1 Прием маскировки под ссылку на авторитет

3.2 Ссылки на неопределенный источник

3.3 Ссылки на свидетельства участников и очевидцев событий, имена и фамилии которых не называются

### 4. Паралогизация

4.1 Ложное причинно-следственное моделирование

4.2 Прием «после этого не значит поэтому»

4.3 Подмена тезиса

4.4 Высказывание о состоянии другого

В дальнейшем будет не важно, какой лингвистический смысл скрывается за каждой из манипуляций, поэтому попробуем ограничиться пониманием того, что из себя представляют четыре группы на верхнем уровне предложенной структуры манипуляций:

- **Негативизация** — группа манипулятивных приемов, приписывающих объекту манипуляции отрицательные качества.
- **Позитивизация** — группа манипулятивных приемов, приписывающих объекту манипуляции позитивные, приукрашенные качества.
- **Деавторизация** — группа манипулятивных приемов, основанных на замалчивании, сокрытии источника информации.
- **Паралогизация** — группа манипулятивных приемов, основанных на отклонении от формальных законов логики.

### 3.3 Подробнее о разметке

Для разметки новостных текстов была привлечена команда лингвистов, а также разработана методика разметки и ее WEB-шаблон в Яндекс Толоке.

#### 3.3.1 Методика разметки

Разметчику подаются новостные тексты, в которых необходимо найти участки текста, где находится манипуляция. Далее, необходимо выбрать один из типов манипуляций и одно слово из списка (будем называть его мишенью манипуляции), на которое эта манипуляция нацелена. Мишени манипуляции являются частью текста, они пронумерованы в нем и выделены следующим образом:



\*(43) русских спецназовца\* — здесь 43 это номер мишени в списке, остальное — название мишени.

Инструкция для разметки:

1. найти все участки текста, где есть манипуляции.
2. для найденной манипуляции обязательно выбрать ее тип и мишень из предложенного списка мишеней.
3. когда выбран тип и мишень манипуляции, необходимо выделить непрерывный фрагмент текста, содержащий саму манипуляцию из пункта 2. Если непрерывный фрагмент выделить не удастся (например, манипуляция разбита на 2 части уже выделенной ранее манипуляцией), то разрешается игнорировать ее.

### 3.3.2 Шаблон разметки

Был разработан шаблон, в котором разметчик курсором мыши выделяет часть текста и присваивает ей тип и мишень манипуляции. Также предусмотрен случай, когда мишени не оказывается в тексте несмотря на наличие манипуляции. У шаблона есть ограничения: так, фрагмент текста, содержащий манипуляцию — непрерывная подстрока и эту подстроку далее невозможно повторно выбрать. По этой причине некоторые сложные случаи, когда в одном и том же фрагменте текста содержатся две манипуляции, либо одна из них вложена в другую, не получится разметить. Однако, согласно проведенному анализу новостных текстов, такие случаи встречаются крайне редко, поэтому ими можно пренебречь без потери информации и качества разметки.

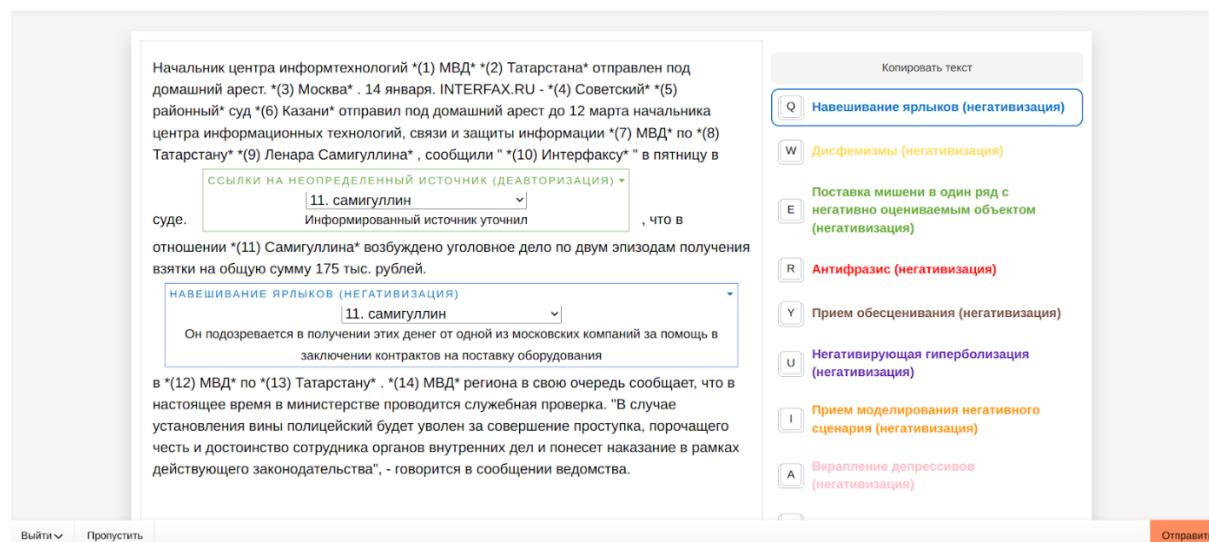


Рис. 3: Скриншот реализованного шаблона разметки в Яндекс Толоке

### 3.4 Анализ размеченной выборки

В случае с разметкой языковых средств манипуляции могут возникать разногласия среди разметчиков, так как процесс проводится согласно лингвистическим представлениям участников, а также опыту их деятельности и работы. Поэтому один и тот же фрагмент текста может быть размечен как две различные, но очень похожие по смыслу манипуляции. Также не всегда можно точно определить мишень манипуляции. Поэтому важно проанализировать характерные особенности полученного набора размеченных новостных документов.

По результатам разметки важно пронаблюдать, как распределены манипуляции, каких из них больше. Эту аналитику можно использовать, чтобы построить более точные модели и в принципе оценить качество проведенной разметки.

#### Число манипуляций:



Рис. 4: Встречаемость всех типов манипуляций

По результатам проведенной работы можно оценить встречаемость манипуляций в текстах на Рис. 4. Видно, что чаще всего в текстах применяется позитивная эвфемизация (подмена исходного слова на позитивно окрашенный аналог), встречаемость остальных типов манипуляции значительно меньше. Также некоторые манипуляции встречаются совсем редко, поэтому существует вероятность переобучения модели на

таких данных. Таким образом, есть доминирующие манипуляции по встречаемости, группа редких манипуляций, которых, вероятно, не будет достаточно для обучения модели, и те манипуляции, которых достаточно в размеченном множестве текстов.

На Рис. 5 манипуляции распределены по четырем группам, которые составляет верхний уровень структуры манипуляций. Такая группировка может быть полезна в том случае, если модели не будет недостаточно обучающих примеров для какого-то типа манипуляций. В результате такой разбивки можно сделать вывод, что в новостных текстах одинаково часто встречаются негативизация и позитивизация, а также деавторизация и паралогизация. Также интересно, что распределение получилось весьма симметричным.

### Число манипуляций в новости:

Новостные тексты в первую очередь должны содержать в себе объективную информацию, поэтому важно проверить, что в целом манипуляции встречаются не слишком часто. На Рис. 6 видно, что примерно в половине размеченных текстов, в которых присутствуют манипуляции, была найдена лишь 1 манипуляция, а тексты с большим количеством манипуляций практически не встречаются. В остальных текстах манипуляции найдены не были, всего же было размечено 3384 текста.

Таким образом, по приведенному на Рис. 6 распределению можно сделать вывод, что число манипуляций на текст обычно не превышает 5-6. А те тексты, где количество манипуляций больше семи, можно считать продуктом крайне манипулятивной (желтой) прессы.

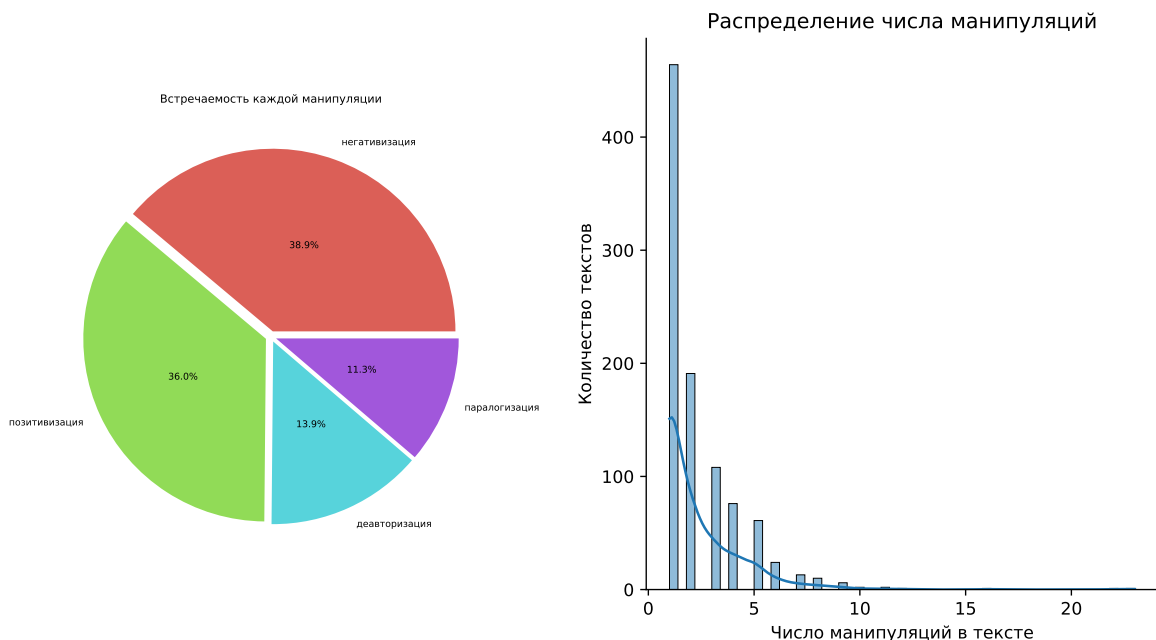


Рис. 5: Встречаемость групп манипуляций

Рис. 6: Распределение числа манипуляций на один новостной текст

### Средняя длина манипуляции:

Так как модель должна будет выделять непрерывные подстроки новостного текста и сопоставлять им типы манипуляций, то важную роль в устойчивости и качестве работы этого алгоритма играет средняя длина манипуляции, которая была получена на разметке. Замерим усреднённую длину в символах всех подстрок, соответствующих одной манипуляции. Результаты на Рис. 7:

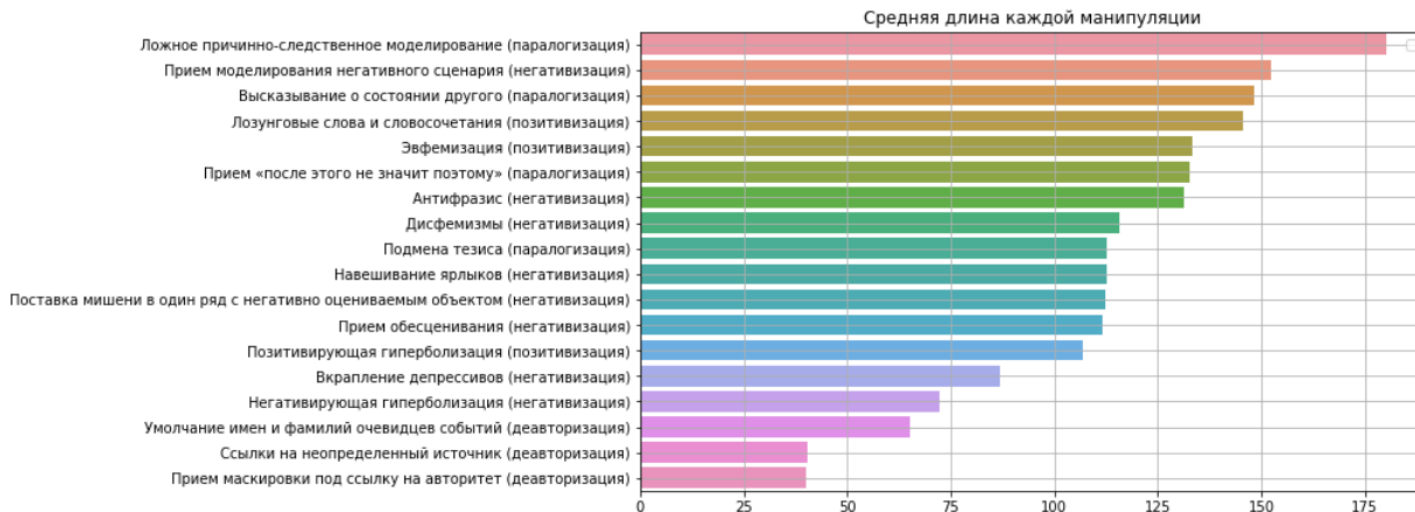


Рис. 7: Средняя длина манипуляций

Интересно отметить, что ссылка на авторитет имеет наименьшую среднюю длину из всех манипуляций, а ложное причинно-следственное моделирование (то есть выстраивание логических цепочек) — наибольшую.

### Согласованность разметки:

Чтобы уменьшить влияние человеческой субъективности на результат разметки, предлагается размечать каждый текст тремя различными разметчиками. Такой подход требует разработки метрики уверенности в разметке текстового документа, по которой можно определить сложность и согласованность разметки выбранного документа.

Для этого был предложен следующий критерий. Он использовался в случае, когда тексты были размечены тремя разметчиками, однако он способна отражать согласованность разметки и при произвольном количестве разметчиков. Введем критерий уверенности разметки и условные обозначения к нему в самом общем виде:

Пусть мы рассматриваем текст  $T$  и его разметило  $K$  разметчиков, то есть  $M_T = \{M_1, M_2, \dots, M_K\}$  есть множество доступных нам разметок от каждого из разметчиков. Напомним, что  $\forall i : M_i = \{(Target\_ID_j, Manip\_ID_j, Text_j) : j \in 1, \dots, N_i\}$ , где:

- $Target\_ID_j$  — ID мишени манипуляции, относящейся к  $j$ -й найденной тройке. Так как мишени получены с помощью NER, то, можно считать, что ID являются общими на всю разметку и с их помощью можно отличить мишени.

- $Manip\_ID_j$  — название манипуляции в  $j$ -й тройке.
- $Text_j$  — фрагмент текста, в котором содержится манипуляция.

Каждая из разметок состоит из троек: типа манипуляции, мишени манипуляции и текста, содержащего манипуляцию. Назовем две тройки с номерами  $j$  и  $k$  совпадающими тогда и только тогда, когда совпадают мишени манипуляции, типы манипуляции и расстояние Левенштейна между фрагментами текста  $Lev(Text_j, Text_k) \leq 6$  (в дальнейшем поясним выбор этого значения). Такая степень близости выбирается для того, чтобы учесть погрешности выделения манипуляций каждым разметчиком. Определим расстояние Левенштейна между двумя текстами через минимальное количество операций замен, вставки или удалений символов, необходимых для превращения одной текстовой строки в другую.

Тогда метрику согласованности разметки можно записать следующим образом:

$$conf(T, M_T) = \frac{\sum_{m_a, m_b \in \{M_1 \cup \dots \cup M_K\}} \mathbb{1}[m_a = m_b]}{K * |M_1 \cup \dots \cup M_K|}$$

Чтобы убедиться в корректности формулы, нужно заметить, что количество троек похожих на не превосходит  $K$ , ведь каждая тройка встречается не более одного раза в разметке. Тогда, просуммировав число похожих троек на каждую тройки из разметки, в числителе будет стоять величина, не превышающая  $K * |M_1 \cup \dots \cup M_K|$ . Отсюда можно заметить, что чем больше уверенности в разметке одного текста, тем ближе эта метрика к 1, и чем меньше совпадений, тем ближе она к 0. Таким образом будет введена согласованность разметки текста.

Измерим уверенность разметчиков по всем текстам. В силу того, что иногда приемы манипуляции плохо отличимы друг от друга и в лингвистике нет математически строгих определений, а также, в силу того, что фрагмент, содержащий манипуляцию, может быть выделен с сильным избытком, будет считать, что тексты, у которых уровень уверенности больше 0.5, хорошо подходят для использования алгоритмами.

Полученный график демонстрирует, что в 51% документов уверенность в разметке максимальная, а при допустимом уровне уверенности  $conf(T, M_T) \geq 0.5$  количество хорошо размеченных документов составляет 85% размеченной выборки, что является хорошим результатом. Далее будем использовать лишь эти тексты.

## 4 Двухуровневая модель

Определимся с форматом проведения экспериментов, а также формализуем все необходимые этапы.

### 4.1 Декомпозиция задачи

Проведем сведение более сложной задачи выделения троек к двум более простым этапам:

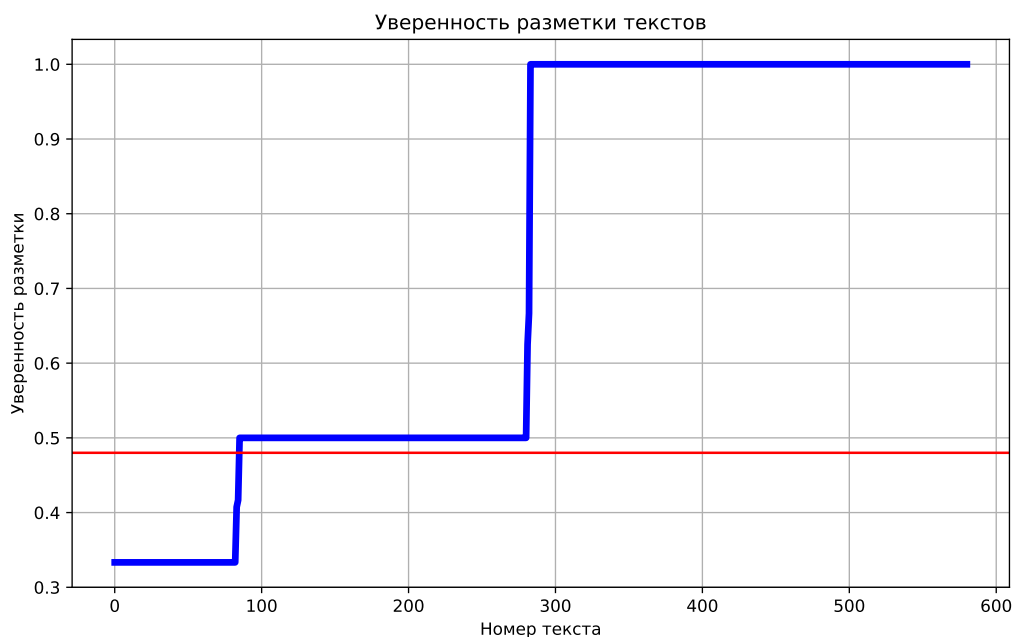


Рис. 8: Уверенность разметки текстов

1. На первом этапе будет создан алгоритм, который сможет выделять фрагменты текста, в которых содержатся манипуляции, а также сами типы манипуляций.
2. На втором этапе будет происходить связывание найденных фрагментов, содержащих языковые манипуляции, с выделенными в тексте мишенями манипуляции.

## 4.2 Обучающее множество

По проведенному анализу размеченного множества можно заметить, что некоторые типы манипуляций встречаются весьма редко, поэтому ожидается, что модели машинного обучения не смогут обучиться достаточно хорошо на таком малом наборе данных. Также важно, чтобы в тестовом множестве присутствовали те же типы манипуляций, что и в обучающем.

Опишем в общем случае алгоритм, с помощью которого составлялись обучающая и тестовая выборки. Пусть имеется  $Z$  типов манипуляций и  $N$  текстов. Тогда любой текст  $\forall n \in \{1 \dots N\}$  содержит в себе  $M_{nz}$  манипуляций типа  $z \in \{1 \dots Z\}$ . Таким образом, каждый текст  $n$  описывается множеством чисел:  $text(n) \equiv \{M_{n1}, M_{n2}, \dots, M_{nZ}\}$ . Обозначим за  $Test$  — множество текстов тестового множества и  $Train$  — множество текстов обучающего множества.

Для задачи первого этапа разобьём каждый текст на предложения. Если текст манипуляции из тройки состоял из нескольких предложений, то сформируем новые

тройки по следующему правилу:

$$(Target\_ID, Manip\_ID, Text = \{Sentence_1, Sentence_2, \dots, Sentence_K\}) \rightarrow \rightarrow \bigcup_{i \in 1 \dots K} (Target\_ID, Manip\_ID, Sentence_i) \quad (1)$$

Отношение числа троек каждого типа манипуляций в Train и Test множествах поддерживается равным 3:1, а те предложения, которые не содержат в себе троек, распределяются случайным образом по той же пропорции.

Для задачи второго этапа модели нужен новостной текст, множество мишеней и список выделенных фрагментов манипуляций, поэтому тексты могут повторяться в Train и Test, а списки выделенных манипуляций отличаться. Тогда, в соответствии с информацией о количестве манипуляций в каждом тексте случайным образом 20% манипуляций каждого типа попадают в Test, а остальные в Train множество.

В тоге каждая модель будет оценена по способности решать следующие три задач:

1. Нахождение троек манипуляций, соответствующих верхнему уровню структуры манипуляций;
2. Нахождение троек из топ-5 самых часто встречающихся манипуляций;
3. Нахождение всех манипуляций.

### 4.3 Оценка качества

Для оценки качества решения главной задачи выделим три критерия:

1. Оценка качества выделения фрагментов текста, содержащих манипуляции. Метрика получается путем усреднения модифицированной метрики Жаккара по всем выделенным фрагментам текста  $X$  по сравнению с размеченными, где  $X = \{x_1, \dots, x_M\}$ .

Таким образом, если алгоритм выделил подстроку  $x$ , а правильные подстроки текста (содержащие манипуляции) — это  $Y = \{Y_1, \dots, Y_N\}$ , то качество выделения фрагмента  $X$  оценивается по формуле ниже:

$$j(x, Y) = \max\left\{\frac{|x \cap Y_1|}{|x \cup Y_1|}, \dots, \frac{|x \cap Y_N|}{|x \cup Y_N|}, 0\right\}$$

Здесь под пересечением подразумевается наибольшая общая подстрока двух строк, под оператором  $|\dots|$  длина строки в символах, а под объединением — конкатенация строк, за исключением, быть может, общей подстроки. То есть "abaa"  $\cap$  "aacd" = "aa" , "abaa"  $\cup$  "aacd" = "aabaacd"

Тогда итоговый критерий записывается путем усреднения модифицированных метрик Жаккара по всем выделенным алгоритмом фрагментам текста:

$$J(X, Y) = \frac{\sum_{i=1}^M j(x_i, Y)}{M}$$

2. Оценка качества классификации типов манипуляции (метрика f1). Рассматриваем лишь те тройки, в которых модифицированная метрика Жаккара  $j$  между подстроками не ниже, чем 0.4 (в дальнейшем поясним выбор этого значения). Напомним, что:

**Точностью (precision)** будем называть долю манипуляций, в которых предсказанный алгоритмом тип манипуляции совпал с размеченным, среди всех манипуляций, найденных алгоритмом.

$$precision = \frac{TP}{TP + FP}$$

**Полнотой (recall)** будем называть долю манипуляций, в которых предсказанный алгоритмом тип манипуляции совпал с размеченным, среди всех исходно размеченных манипуляций.

$$recall = \frac{TP}{TP + FN}$$

Их усреднение, объединяющее точность и полноту, называется f1-мерой. Чем она ближе к единице, тем выше качество классификации типов манипуляции. f1-мера задается:

$$f1 = \frac{2 \times precision \times recall}{precision + recall}$$

3. Оценка качества связывания фрагментов манипуляции с выделенной сущностью в тексте (метрика f1). Рассматриваются лишь те тройки, которые считались позитивными в предыдущем примере. Если фрагмент манипуляции не выделен или выделен неверно, то не учитываем его в подсчете. Остальное аналогично:

**Точностью (precision)** будем называть долю пар фрагмент текста/сущность, в которых предсказанный алгоритмом результат совпал с размеченным, среди пар, найденных алгоритмом.

**Полнотой (recall)** будем называть долю пар фрагмент текста/сущность, в которых предсказанный алгоритмом результат совпал с размеченным, среди всех исходно размеченных таких пар.

## 4.4 Описание экспериментов

В этой части работы будут продемонстрированы подходы к обучению различных нейросетевых моделей машинного обучения для решения задач первого и второго этапа, а так же модификации представленных методов. Будет измерено качество выделения фрагментов текста с манипуляцией, сопоставление им типов и мишеней манипуляции.



### 4.4.1 NER и RE

В качестве модели первого этапа выберем предобученный NER из библиотеки DeepPavlov. Как и описывалось ранее, будем его дообучать на Train, то есть, результатом работы модели станет последовательность символов, равное количество входных слов. Каждый из символов обозначает, является ли слово частью манипуляции и если да, то какого типа.

В качестве модели второго типа будет использована предобученный RE [15] из библиотеки DeepPavlov. И аналогично на обучающем множестве второго этапа будет производиться дообучение.

#### Как устроен RE:

Задаче RE (Relation Extraction) заключается в том, чтобы по контексту определить, связаны ли друг с другом сущности (например, из NER) и если да, то какой у них тип связи. В нашем случае упростим задачи до бинарного типа связи, то есть: либо связь есть, либо она отсутствует. С помощью этого алгоритма будет установлено, существует ли связь между потенциальной мишенью манипуляции и фрагментом текста, содержащим эту манипуляцию. Выбранная модель получает на вход текст, в котором специальными символами отмечены сущности. Далее используется предобученный BERT [16] для получения внутреннего представления токенов:

$$H = [h_1, h_2, \dots, h_l] = BERT([x_1, x_2, \dots, x_l]),$$

где исходный текст  $T = [x_i]_{i=1}^l$

Далее векторные представления токенов, из которых состоит сущность, агрегируются в числовой вектор. Таким образом, из векторов сущностей с помощью линейных преобразований и функции сигмоиды вычисляется вероятность того, что между парами сущностей имеется связь.

$$\begin{aligned} z_i &= \tanh(W_s \hat{h}_a), \\ z_j &= \tanh(W_s \hat{h}_b), \\ Prob(r|a, b) &= \sigma(z_i^T W_r z_j + b_r), \\ \sigma(x) &= \frac{1}{1 + \exp(-x)}, \end{aligned}$$

где  $a$  и  $b$  означают две сущности, между которыми высчитывается вероятность существования связи,  $\hat{h}_a$  и  $\hat{h}_b$  — их векторные представления после агрегации. Остальное — обучаемые параметры.

Важной особенностью метода является обучаемое значение той вероятности, при котором можно утверждать наличие связи. В итоге, действуя как в статье и дообучая BERT и сопутствующие параметры на наших данных, получаем итоговую модель.

### 4.4.2 NER\_BERT, NER\_BERT+LL

Так как алгоритм с использованием Bi-LSTM [12] (базовый вариант NER), является менее актуальным подходом к задачам моделирования языка, то в качестве

модификации модели первого этапа будет попытка использовать трансформер. Более формально текст  $T = [x_t]_{t=1}^l$ , как и ранее, преобразовывается в  $H = [h_1, h_2, \dots, h_l] = BERT([x_1, x_2, \dots, x_l])$  с помощью одного из предобученных заранее на русскоязычных текстах моделей-трансформеров. Далее, с помощью линейного преобразования каждому токenu сопоставляется распределение вероятности принадлежности к типам манипуляции, согласно которому токenu назначается тип манипуляции, либо ее отсутствие.

В качестве предобученного трансформера рассматривались XLMRoberta[17] и RuBERT [18]. Так же одним из вариантов эксперимента является дообучение всей модели (NER\_BERT) или лишь последнего линейного преобразования (NER\_BERT+LL).

### 4.4.3 RE, RE+LL

Аналогично, в модели RE можно обучать всю модель или только финальные линейные преобразования. В силу того, что в собранном и размеченном наборе данных некоторых манипуляций может быть недостаточно для возможности хорошо находить эту манипуляцию в тексте, обучение лишь последнего слоя выглядит более перспективной модификацией, позволяющей избежать эффекта переобучения модели под обучающую выборку.

## 4.5 Результаты

В результатах обучим и измерим качество моделей по трем метрикам по трем задачам (ALL — модель обучается искать все манипуляции, TOP-5 — модель обучается искать пять самых часто встречающихся манипуляций, CATEG — модель учится искать манипуляции с верхнего уровня разработанной структуры манипуляций). Также проведем эксперименты с полным перебором тех моделей, которые были описаны выше. Добавка + LL означает, что обучалась не вся модель целиком, а лишь ее лишь ее последние слои в соответствии с тем, как было рассказано выше.

	ALL	TOP-5	CATEG
JACC	0.540	0.563	0.611
$f1_{match}$	0.459	0.482	0.503
$f1_{manip}$	0.476	0.513	0.547

Таблица 1: NER + RE

	ALL	TOP-5	CATEG
JACC	0.321	0.378	0.428
$f1_{match}$	0.499	0.533	0.687
$f1_{manip}$	0.252	0.295	0.323

Таблица 2: NER\_rubert + RE

По результатам можно заметить, что чем больше в модели обучаемых параметров и чем она сложнее, тем при текущем объеме размеченных данных хуже работает алгоритм. Если сравнивать результаты NER\_rubert и NER\_xlmroberta с их модификациями, когда обучаются лишь линейные преобразования, то разница в качестве поиска манипуляций ощутима. Однако, так как модель первого уровня находит меньше манипуляций, то в модели второго уровня приходит меньше манипуляций. Ско-

	ALL	TOP-5	CATEG
JACC	0.564	0.579	0.636
$f1_{match}$	0.460	0.484	0.515
$f1_{manip}$	0.483	0.515	0.548

Таблица 3: NER\_rubert + LL + RE

	ALL	TOP-5	CATEG
JACC	0.303	0.349	0.396
$f1_{match}$	0.478	0.527	0.630
$f1_{manip}$	0.223	0.241	0.301

Таблица 4: NER\_xlmroberta + RE

	ALL	TOP-5	CATEG
JACC	0.544	0.557	0.604
$f1_{match}$	0.472	0.493	0.532
$f1_{manip}$	0.457	0.502	0.544

Таблица 5: NER\_xlmroberta + LL + RE

	ALL	TOP-5	CATEG
JACC	0.540	0.563	0.611
$f1_{match}$	0.483	0.501	0.558
$f1_{manip}$	0.476	0.513	0.547

Таблица 6: NER + RE + LL

	ALL	TOP-5	CATEG
JACC	0.321	0.378	0.428
$f1_{match}$	0.500	0.523	0.663
$f1_{manip}$	0.252	0.295	0.323

Таблица 7: NER\_rubert+ RE + LL

	ALL	TOP-5	CATEG
JACC	0.564	0.579	0.636
$f1_{match}$	0.463	0.489	0.521
$f1_{manip}$	0.483	0.515	0.548

Таблица 8: NER\_rubert + LL + RE + LL

	ALL	TOP-5	CATEG
JACC	0.303	0.349	0.396
$f1_{match}$	0.481	0.513	0.624
$f1_{manip}$	0.223	0.241	0.301

Таблица 9: NER\_xlmroberta + RE + LL

	ALL	TOP-5	CATEG
JACC	0.544	0.557	0.604
$f1_{match}$	0.478	0.494	0.527
$f1_{manip}$	0.457	0.502	0.544

Таблица 10: NER\_xlmroberta + LL + RE + LL

рее всего, это совсем простые по структуре строковые объекты, связывать которые с необходимыми сущностями намного проще. Поэтому мера качества связывания манипуляций с мишенями намного выше во всех экспериментах при плохо работающей модели первого уровня.

Также использование в обучении модели RE лишь финальных линейных преобразований не всегда позволяет поднять качество модели второго уровня. Так, в Таблицах 10 и 5 видно, что упрощение модели второго уровня приводит к некоторой потере качества в задаче связывания найденных манипуляций и потенциальной мишеней манипуляции. Скорее всего это связано с тем, что мультиязычная модель хуже улавливает языковые особенности русского языка и манипуляции, которые ей удастся найти, имеют более "общий" характер. То есть, текстовая структура этих манипуляций не позволяет дать точный ответ на то, на какую мишень она была направлена.

Если сравнивать эффективность использования мультиязычной XLMRobert-ы и RuBERT-a, обученного на русскоязычных текстах, то ожидаемо второй алгоритм

позволяет достичь большего качества в задаче поиске и классификации манипуляций.

По итогу решения трех задач ALL, TOP-5 и CATEG подтверждается гипотеза о том, что для первых двух возникает нехватка обучающих данных. В первой задаче требуется выделять все манипуляции, поэтому модель переобучается в типах манипуляций, которых всего несколько процентов от общего объема. Во второй задаче модель обучается определять лишь те манипуляции, число которых можно считать достаточным в Train наборе данных. Ожидаемо качество выделения этих манипуляций и качество классификации типов манипуляции выше, чем у первой задаче практически во всех экспериментах. Это говорит о том, что модели первого и второго уровня лучше выучивают те закономерности, которые определяют эти манипуляции. Последняя задача ожидаемо была решена с наилучшими результатами, так как в ней модель обучается находить не сами манипуляции, а группы манипуляций. В этой задаче у каждой группы больше данных для обучения, чем у манипуляций из предыдущих двух задач, а также сильнее наблюдаются отличия между теми типами манипуляций, которые необходимо определить. Упрощенная логика, по которой манипуляции можно отличить друг от друга, а также увеличенный объем данных для обучения позволяет решать задачу CATEG намного лучше остальных.

Обобщая сказанное, можно сделать вывод, что лучший результат продемонстрирован в эксперименте из Таблицы 8. Итоговая двухуровневая модель лучше прочих находит фрагменты текста, в которых содержатся манипуляции, и лучше всех других моделей классифицирует их.

## 5 Модификация модели второго уровня

Исходя из первой серии экспериментов при небольшом объеме обучающей выборки успешно выделить из текстов манипуляции и классифицировать их. Однако задача сопоставления потенциальных мишеней манипуляции с текстами может иметь потенциально более простой способ, чем модель RE. Действительно, для некоторых манипулятивных конструкций важным фактором является контекст, благодаря которому можно определить, относится к ним выбранная мишень или нет. Однако попробуем упростить эту модель.

Ключевое упрощение заключается в том, что вся необходимая информация для сопоставления манипуляций и их мишеней содержится в тексте самой манипуляции и, быть может, в некотором контексте, окружающем мишень манипуляции.

В качестве контекста для мишеней манипуляции возьмем слово левее, если оно вместе с мишенью находится в одном предложении. По аналогичной логике добавим в контекст мишени слово правее. Таким образом, задачу второго этапа можно свести к задаче Dense Retrieval (DR) [19] — задаче ранжирования. Поэтому, изменим подход к обучению второго этапа предложенной двухуровневой модели.

Обучающее множество оставим таким же, как в предыдущем пункте. Для оценивания качества работы алгоритма второго уровня будем использовать критерий

качества  $f1_{manip}$ .

## 5.1 Описание экспериментов

В задаче ранжирования имеется корпус запросов  $Q=queries$  и корпус документов  $D=docs$ . Каждому запросу требуется найти наиболее релевантные документы. Требуется оценить следующее выражение:

$$P(Relevant = 1|d_i, q) \hat{=} \Phi(\eta_Q(q), \eta_D(d))$$

, где  $q$  и  $d$  пара запрос-документ,  $\eta_Q(q)$  и  $\eta_D(d)$  их векторные представления, полученные с помощью некоторых алгоритмов, а  $\Phi$  — мера близости этих векторов. Существует множество подходов к решению этой задачи [20][21][22][23], однако предлагается составить собственный алгоритм. В качестве  $Q$  будем использовать выделенные фрагменты текста с манипуляциями. В качестве  $D$  — кандидаты в мишени манипуляции. Поэтому задача ранжирования становится эквивалентна задаче второго этапа: для каждой манипуляции необходимо найти наиболее релевантную ей мишень манипуляции. Или, наоборот, для каждой мишени найти свой фрагмент текста с манипуляцией. Будут проведены исследования для обоих подходов.

Введем следующие понятия:

- **Позитивы** — набор пар  $P = \{(Text_{manip}^i, Text_{target}^i)^+\}_{i=1}^K$ . Это пары, в которых разметчики установили связь между текстом манипуляции и мишенью манипуляции. Всего таких пар можно выделить  $K$  штук из размеченного множества текстов.

В том случае, когда мишень манипуляции находится вне текста, не считаем такую пару позитивом.

- **Негативы** — набор пар  $N = \{(Text_{any}^i, Text_{target}^i)^-\}_{i=1}^M$ . Это пары, в которых связь между текстом манипуляции и мишенью отсутствует. В дальнейшем набор негативов будет генерироваться для увеличения объема обучающей выборки.

В качестве  $\eta_Q(q)$  и  $\eta_D(d)$  будем использовать 2 трансформера: dual-encoder и cross-encoder.

### 5.1.1 Dual-encoder

На вход модели подаётся токенизированный текст с манипуляцией и текст с мишенью и контекстом. Далее, с помощью кодировщика, в основе которого лежит RuBERT из предыдущих экспериментов, эти тексты преобразуются в числовые векторы и с помощью косинусной функции расстояния вычисляется мера близости между манипуляцией и мишенью.

В качестве  $\Phi$ , меры близости векторов  $x$  и  $y$ , введем косинусную меру близости, которая определяется как:

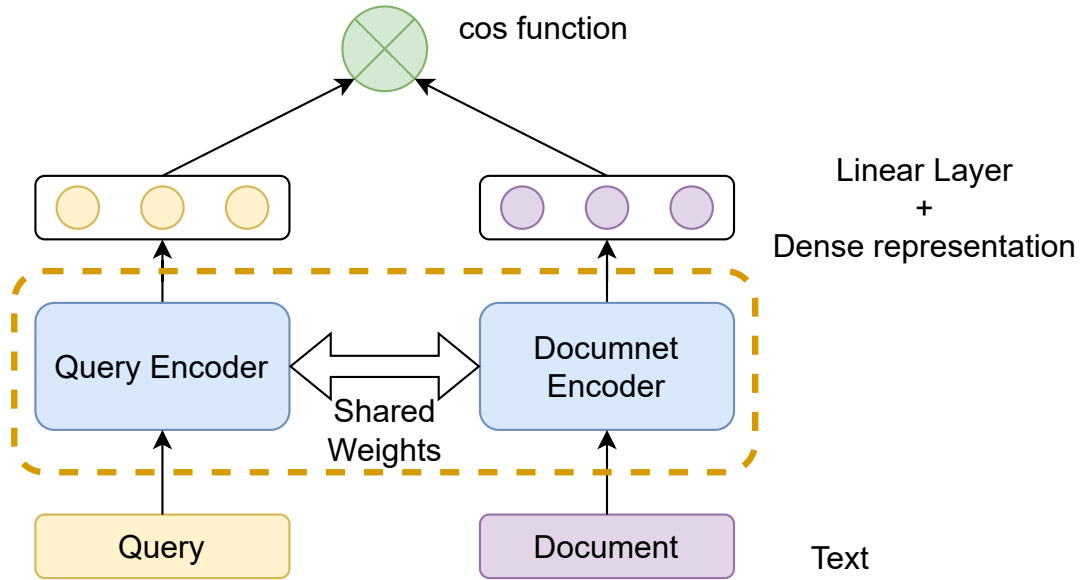


Рис. 9: Схематичное устройство применяемой модели dual-encoder

$$\cos(x, y) = \frac{\sum_{i=1}^n y_i \times x_i}{\sqrt{\sum_{i=1}^n x_i^2} \times \sqrt{\sum_{i=1}^n y_i^2}}$$

Ее значения лежат в диапазоне  $[-1, 1]$ , и чем более похожи векторы, тем косинусная мера ближе к 1.

### 5.1.2 Cross-encoder

Новая модель отличается от предыдущей тем, что способна учитывать близость объектов более точно, так как манипуляция и мишень в ней обрабатываются одновременно. Достигается это с помощью конкатенации [24]:

$$input = ([CLS] \circ [q_0] \circ \dots \circ [q_{N(q)}] \circ [SEP] \circ [d_0] \circ \dots \circ [d_{N(d)}] \circ [SEP]),$$

где  $[SEP]$  и  $[CLS]$  специальные токены, а запрос и документ кодируются токенами  $[q_i]$  и  $[d_i]$  соответственно.

С помощью линейного преобразования и сигмоидной функции получается оценка — мера близости. Таким образом, значения оценки лежат в диапазоне  $[0, 1]$ , и чем более похожи векторы, тем мера ближе к 1.

### 5.1.3 In-batch техника

Этот способ негативного сэмплирования [25] заключается в следующем: случайным образом выбираем  $K$  пар позитивов:

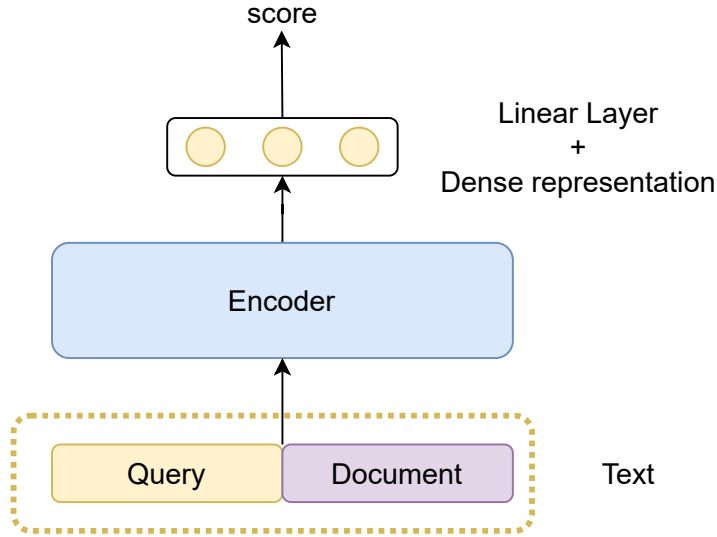


Рис. 10: Схематичное устройство применяемой модели cross-encoder

$$P = \begin{cases} (Text_{manip}^{i_1}, Text_{target}^{i_1})^+ & \text{первый позитив} \\ \dots & \\ (Text_{manip}^{i_K}, Text_{target}^{i_K})^+ & \text{K-й позитив} \end{cases}$$

Тогда, если негативы строятся для манипуляций, то их набор создаётся так:

$$N = \begin{cases} \{Text_{target}^{i_k}\}_{\forall i_k \neq i_1} & \text{негативы для } Text_{manip}^{i_1} \\ \{Text_{target}^{i_k}\}_{\forall i_k \neq i_2} & \text{негативы для } Text_{manip}^{i_2} \\ \dots & \\ \{Text_{target}^{i_k}\}_{\forall i_k \neq i_K} & \text{негативы для } Text_{manip}^{i_K} \end{cases}$$

Аналогичная логика применяется для построения негативов по мишеням манипуляций. То есть, для каждой мишени манипуляции составляется список не связанных с этой мишенью фрагментов текста, содержащих какую-либо манипуляцию.

#### 5.1.4 Негативное сэмплирование и функция потерь

Рассмотрим несколько техник негативного сэмплирования. Негативы по манипуляции составляются из:

1. всех мишеней из одного текста, у которых в разметке не установлена связь с выбранной манипуляцией;
2. всех мишеней из всего размеченного набора текстов, у которых в разметке не установлена связь с выбранной манипуляцией;

### 3. In-batch техника.

Негативы по мишени составляются из:

1. всех манипуляций из одного текста, у которых в разметке не установлена связь с выбранной мишенью;
2. всех манипуляций из всего размеченного набора текстов, у которых в разметке не установлена связь с выбранной мишенью;

### 3. In-batch техника.

## Функция потерь

Основываясь на результатах предыдущего эксперимента, в качестве кодировщика выбирается RuBERT, а обучаются в модели только линейные преобразования. Тогда для выбранного запроса  $q$  (в терминах нашей задачи манипуляции/мишени), позитивного документа  $d^+$  и множества негативов  $N(q)$  (в терминах нашей задачи мишеней/манипуляций) оптимизируется следующий функционал:

$$\mathcal{L} = -\log \frac{e^{\text{sim}(q, d^+)}}{e^{\text{sim}(q, d^+)} + \sum_{d^- \in N(q)} e^{\text{sim}(q, d^-)}},$$

где  $\text{sim}(q, d)$  — мера близости  $q$  и  $d$ .

## 5.2 Результаты

Будем измерять  $f1_{\text{match}}$  для результатов с первого этапа, полученных из эксперимента из Таблицы 8.

### Результаты для dual-encoder:

	ALL	TOP-5	CATEG
baseline	0.463	0.489	0.521
in-batch	0.398	0.447	0.507
in-text	0.389	0.441	0.493
out-text	0.415	0.460	0.535

Таблица 11: Мишень  $\rightarrow$  фрагменты манипуляции

	ALL	TOP-5	CATEG
baseline	0.463	0.489	0.521
in-batch	0.392	0.443	0.499
in-text	0.385	0.436	0.481
out-text	0.406	0.449	0.523

Таблица 12: Фрагменты манипуляции  $\rightarrow$  мишень

### Результаты для cross-encoder:

По итогам экспериментов можно сделать вывод, что задачи сопоставления по мишени фрагментов текста, содержащих манипуляции, и сопоставления по манипуляции ей мишеней очень похожи, однако последняя решается немного хуже первой. Это может быть связано с тем, что в задачах ранжирования обычно запрос намного короче, чем документ. Также в сэмплировании негативов модель узнает больше новой информации из длинных подстрок, содержащих манипуляции, чем из коротких подстрок, содержащих мишени.



	ALL	TOP-5	CATEG
baseline	0.463	0.489	0.521
in-batch	0.459	0.475	0.522
in-text	0.443	0.468	0.511
out-text	0.466	0.484	0.557

Таблица 13: Мишень  $\rightarrow$  фрагменты манипуляции

	ALL	TOP-5	CATEG
baseline	0.463	0.489	0.521
in-batch	0.450	0.466	0.513
in-text	0.437	0.463	0.504
out-text	0.461	0.478	0.531

Таблица 14: Фрагменты манипуляции  $\rightarrow$  мишень

Во всех экспериментах на задачах ALL и TOP-5 не удалось получить результаты лучше, чем baseline. Это может быть связано с тем, что выделение конкретного типа манипуляций сужает вариативность предложений, которые поступают на оценку модели второго этапа, в то время как модель была обучена на более вариативных данных. Из-за отличий поступающих и обучающих данных произошло снижение качества. Это объясняет то, что в экспериментах по задаче CATEG модель второго этапа показала наилучшие результаты. Также, техники сэмплирования негативов, отличаются многообразием того, как много негативов может увидеть модель. Техника out-text позволяет обработать наибольшее число негативов, поэтому в случае применения этой техники модель работает лучше всего.

Как и ожидалось, cross-encoder модель лучше справляется с задачей сопоставления текстов, чем dual-encoder. Это связано с тем, в оценивании меры близости манипуляции и мишени принимает участие вся модель, а не только косинусная мера близости.

## 6 Выбор гиперпараметров

Для отбора качественно размеченных текстов использовалось значение расстояние Левенштейна 6, а при введении критерия качества жаккара значение 0.4. Напомним, что:

- Чем больше расстояние Левенштейна, тем больше текстов считаются качественно размеченными. Таким образом, этот параметр влияет на размер обучающей выборки и строгость отбора текстов в нее.
- От метрики Жаккара зависит, какое количество и качество манипуляций будут переданы второму этапу двухуровневой модели.

Проведем эксперименты на лучшей полученной модели на задаче CATEG:

Можно сделать вывод, что при значении расстояния Левенштейна равном 1, в обучающую выборку попадает недостаточно объектов и модель показывает низкое качество. При увеличении этого параметра до 14 символов ключевые критерии остаются на том же уровне. Поэтому выбор числа 6 оптимален.

Аналогично с метрикой Жаккара, если параметр слишком сильно понизить, то много плохо совпадающих объектов пройдет проверку и качество упадет. А если вы-

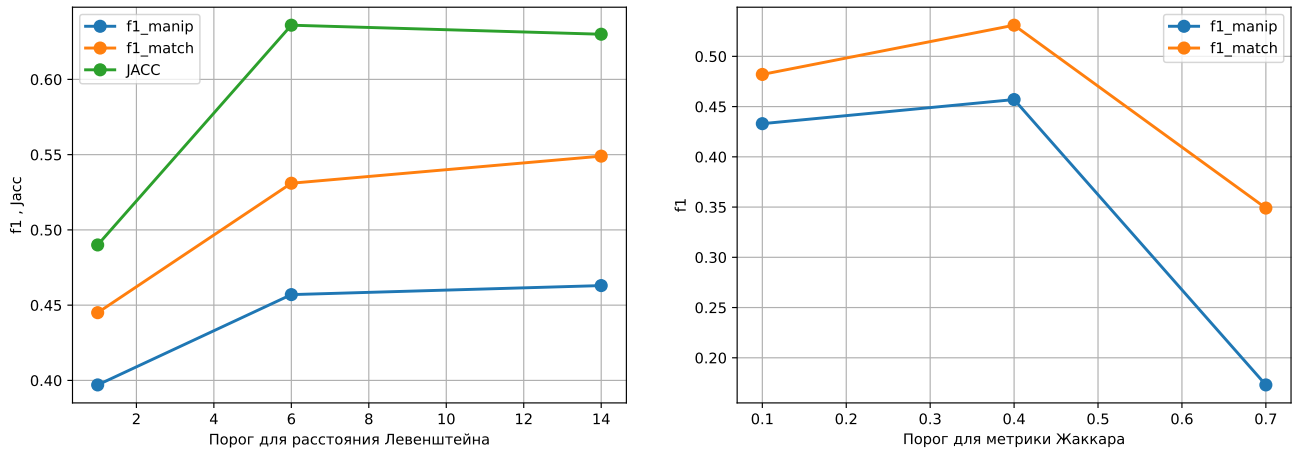


Рис. 11: Эксперименты с гиперпараметрами

брать ее чрезмерно большой, то в оценивание и ко второй модели попадет очень мало объектов, что негативно отражается на качестве. Отсюда выбор числа 0.4 оптимален.

## 7 Заключение

В ходе данной работы было достигнуто несколько важных результатов.

Во-первых, была предложена методика разметки, разработана инструкция разметки и шаблон на Яндекс Толоке, собрана выборка из нескольких тысяч новостных текстов. На ее основе был осуществлен анализ новостного потока, в результате которого была разработана уникальная двухуровневая система манипуляций. Основываясь на ней, была произведена разметка новостных текстов и получен датасет с манипуляциями с долей качественной разметки текстов — 85%.

Во-вторых, была произведена декомпозиция исходной задачи на выделение манипуляций и сопоставление выделенным манипуляциям мишеней манипуляции, а также предложено несколько упрощений исходной задачи классификации манипуляций. Была разработана методика обучения моделей для задач первого и второго уровня декомпозиции, а также критерии, позволяющие оценивать качество выделения манипуляций. В результате проведения множества экспериментов и перебора подходов к решению задач обоих этапов, удалось найти оптимальный алгоритм, обеспечивающий достойное качество выделения манипуляций.

## Список литературы

- [1] Uscinski J. E., Butler R. W. The epistemology of fact checking //Critical Review. – 2013. – Т. 25. – №. 2. – С. 162-180.
- [2] Graves L., Cherubini F. The rise of fact-checking sites in Europe. – 2016.
- [3] Zubiaga A. et al. Detection and resolution of rumours in social media: A survey //ACM Computing Surveys (CSUR). – 2018. – Т. 51. – №. 2. – С. 1-36.
- [4] Rashkin H. et al. Truth of varying shades: Analyzing language in fake news and political fact-checking //Proceedings of the 2017 conference on empirical methods in natural language processing. – 2017. – С. 2931-2937.
- [5] Baly R. et al. Integrating stance detection and fact checking in a unified corpus //arXiv preprint arXiv:1804.08012. – 2018.
- [6] Dori-Hacohen S., Allan J. Automated controversy detection on the web //European Conference on Information Retrieval. – Springer, Cham, 2015. – С. 423-434.
- [7] Ciampaglia G. L. et al. Computational fact checking from knowledge networks //PloS one. – 2015. – Т. 10. – №. 6. – С. e0128193
- [8] Dai J., Xu J. Knowledge Graph Construction for Intelligent Media Based on Mobile Internet //Wireless Communications & Mobile Computing. – 2022. – Т. 2022.
- [9] Lê T. A., Burtsev M. S. A deep neural network model for the task of Named Entity Recognition //International Journal of Machine Learning and Computing. – 2019. – Т. 9. – №. 1. – С. 8-13.
- [10] Zhang X., Zhao J., LeCun Y. Character-level convolutional networks for text classification //Advances in neural information processing systems. – 2015. – Т. 28.
- [11] Kim Y. et al. Character-aware neural language models //Thirtieth AAAI conference on artificial intelligence. – 2016.
- [12] Hochreiter S., Schmidhuber J. Long short-term memory //Neural computation. – 1997. – Т. 9. – №. 8. – С. 1735-1780.
- [13] Sutton C. et al. An introduction to conditional random fields //Foundations and Trends in Machine Learning. – 2012. – Т. 4. – №. 4. – С. 267-373.
- [14] Joulin A. et al. Bag of tricks for efficient text classification //arXiv preprint arXiv:1607.01759. – 2016.
- [15] Zhou W. et al. Document-level relation extraction with adaptive thresholding and localized context pooling //Proceedings of the AAAI Conference on Artificial Intelligence. – 2021. – Т. 35. – №. 16. – С. 14612-14620.

- [16] Wolf T. et al. Huggingface’s transformers: State-of-the-art natural language processing //arXiv preprint arXiv:1910.03771. – 2019.
- [17] Conneau A. et al. Unsupervised cross-lingual representation learning at scale //arXiv preprint arXiv:1911.02116. – 2019.
- [18] Kuratov Y., Arkhipov M. Adaptation of deep bidirectional multilingual transformers for russian language //arXiv preprint arXiv:1905.07213. – 2019.
- [19] Zhan J. et al. Learning to retrieve: How to train a dense retrieval model effectively and efficiently //arXiv preprint arXiv:2010.10469. – 2020.
- [20] Guo J. et al. Semantic models for the first-stage retrieval: A comprehensive review //ACM Transactions on Information Systems (TOIS). – 2022. – T. 40. – №. 4. – C. 1-42.
- [21] Lewis P. et al. Boosted Dense Retriever //arXiv preprint arXiv:2112.07771. – 2021.
- [22] Lu S. et al. Less is More: Pre-train a Strong Text Encoder for Dense Retrieval Using a Weak Decoder //arXiv preprint arXiv:2102.09206. – 2021.
- [23] Xu C. et al. LaPraDoR: Unsupervised Pretrained Dense Retriever for Zero-Shot Text Retrieval //arXiv preprint arXiv:2203.06169. – 2022.
- [24] Yu H. C., Xiong C., Callan J. Improving query representations for dense retrieval with pseudo relevance feedback //arXiv preprint arXiv:2108.13454. – 2021.
- [25] Qu Y. et al. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering //arXiv preprint arXiv:2010.08191. – 2020.