

РЕФЕРАТ

Расчетно-пояснительная записка 57 с., 17 рис., 3 таблиц, 33 источника, 2 приложения.
ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ, ГЕНЕРАЦИЯ ПОДПИСЕЙ, КОМПЬЮТЕРНОЕ ЗРЕНИЕ, ОБРАБОТКА ЕСТЕСТВЕННОГО ЯЗЫКА, ЮМОР, МАШИННОЕ ОБУЧЕНИЕ. Цель работы — разработка программного решения, использующего искусственный интеллект для генерации смешных подписей к изображениям. В рамках курсовой работы был проведен анализ предметной области, связанной с юмором, его видами и стилями, а также обзор существующих инструментов для создания подписей. Были сформулированы требования к проектируемому программному обеспечению. Изучены и описаны схемы алгоритмов контекстуальной обработки изображений, генерации текстовых подсказок (промтov) и непосредственно генерации смешных подписей с использованием современных моделей машинного обучения. Разработанный программный комплекс основан на использовании передовых нейросетевых моделей для анализа визуальной информации и последующей генерации релевантного и юмористического текста. Описаны выбранные средства реализации, включая язык программирования и ключевые библиотеки. Проведены тестовые испытания, позволяющие оценить производительность разработанного решения.

Содержание

РЕФЕРАТ	4
ВВЕДЕНИЕ	8
1 Аналитическая часть	9
1.1 Картинки и смешные подписи к ним	9
1.2 Смех и смешное	9
1.3 Стили юмора	10
1.4 Виды шуток	10
1.5 Сфера применения	11
1.6 Постановка задачи	11
1.7 Обзор существующих аналогов	12
1.7.1 Supermeme.ai	12
1.7.2 Imgflip	12
1.7.3 Pincel	12
1.7.4 Magic Hour	12
1.7.5 Funny caption generator	13
1.7.6 Pallyy	13
1.7.7 Aigo.tools	13
1.7.8 Galaxy AI Image Caption Generator	13
1.7.9 WritingMate AI Image Caption Generator	14
1.7.10 Gemini	14
1.7.11 Neural Joking Machine	14
1.7.12 Oxford Humorous Image Captions	14
1.7.13 Humor in AI: Massive Scale Crowd-Sourced Preferences and Benchmarks for Cartoon Captioning	15
1.7.14 AI Humor Generation: Cognitive, Social and Creative Skills for Effective Humor	15
1.7.15 XMeCap: Meme Caption Generation with Sub-Image Adaptability	15
1.7.16 Meme – Generator	16
1.7.17 contxtual_cuptions	16

1.7.18	DeepSeek AI	16
1.7.19	ChatGPT	16
1.7.20	Show and Tell: A Neural Image Caption Generator	17
1.7.21	Image Caption Generator	17
1.7.22	Junia AI Image Caption Generator	17
1.7.23	10015.io Image Caption Generator	17
1.7.24	ZapCap Funny Instagram Caption Generator	18
1.7.25	Chat100 Photo Caption Generator	18
1.7.26	Simplified	18
1.7.27	Captionslab (aigo.tools)	18
1.7.28	YesChat Image Caption Generator	18
1.7.29	Thread Creator Image Caption Generator	18
1.7.30	Videotok Caption Generator	19
1.8	Сравнение существующих аналогов	19
1.9	Ограничения и этические аспекты практической реализации	22
2	Конструкторская часть	25
2.1	Общая схема метода	25
2.2	Описание алгоритмов системы	26
2.2.1	Контекстуальная обработка изображения	26
2.2.2	Генерация промтта	30
2.2.3	Генерация смешной подписи	31
2.3	Сценарий взаимодействия пользователя с программой	34
3	Технологическая часть	36
3.1	Язык программирования и среда разработки	36
3.2	Выбор средств реализации программного обеспечения	36
3.3	Компоненты программного обеспечения	38
3.3.1	Инициализация моделей	38
3.3.2	Контекстуальное описание изображения	38
3.3.3	Перевод описания на русский	39
3.3.4	Генерация смешной подписи	40
3.3.5	Взаимодействие с пользователем	41
3.3.6	Установка и запуск	41
3.3.7	Физические компоненты системы	42
4	Исследовательская часть	43
4.1	Характеристики тестового стенда	43
4.2	Методика оценки производительности и потребления ресурсов	43
4.3	Данные по тестируемым изображениям	44

5 Организационно-правовая часть	51
5.1 Нейронные сети и искусственный интеллект в контексте российского законодательства	51
5.2 Проблемы цензурирования и ограничения практического использования ПО	51
5.3 Лицензионная часть	52
5.4 Выбор лицензии для разрабатываемого ПО	54
ЗАКЛЮЧЕНИЕ	56
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	57
ПРИЛОЖЕНИЕ А	60
ПРИЛОЖЕНИЕ Б	64

ВВЕДЕНИЕ

Автоматическая генерация смешных подписей к изображениям – это технология, позволяющая автоматически создавать юмористический текстовый контент для визуальных материалов. В условиях современного цифрового мира, где визуальный контент играет ключевую роль в коммуникации, способность быстро и эффективно создавать привлекательные и, в частности, юмористические подписи к изображениям приобретает все большее значение.

Целью данной курсовой работы является разработка программного решения, использующего искусственный интеллект для генерации смешных подписей к изображениям. Чтобы достигнуть поставленной цели, требуется решить следующие задачи.

- 1) Анализ существующих решений для генерации подписей к изображениям, включая их функциональность и используемые подходы.
- 2) Выбор и обоснование методов искусственного интеллекта для реализации системы генерации юмористических подписей.
- 3) Разработка и тестирование программного обеспечения, способного генерировать смешные подписи к изображениям.
- 4) Исследование времени работы программного решения и его затрат памяти.
- 5) Исследование организационно-правовой составляющей программного решения и защита его лицензией.

1 Аналитическая часть

1.1 Картинки и смешные подписи к ним

Картинки со смешными подписями к ним приятно называть мемами. Мем – единица культурной информации, любая идея, символ, манера или образ действия, осознанно или неосознанно передаваемые от человека к человеку посредством речи, письма, видео, ритуалов, жестов и т. д. [1] Так как картинка со смешной подписью к ней является мемом, то дальше будет использоваться именно это слово.

Ключевые характеристики мемов:

- мем должен быть легко воспроизводимым и распространяться среди большого количества людей.
- мемы часто подвергаются изменениям и адаптациям со стороны разных пользователей.
- мемы отражают определенные культурные тренды, шутки, события или социальные явления.
- мемы часто используются для создания комического эффекта, но они могут также нести сатирический, ироничный, эмоциональный или даже серьезный характер.

Так как мемы направлены на большую аудиторию, они часто распространяются в виде нейтральных картинок со смешными подписями, которые понятны большей части людей и используют универсальные темы, ситуации или архетипы, которые могут быть интерпретированы и оценены людьми с разным культурным бэкграундом, возрастом и интересами. Однако существуют виды мемов, которые направлены на узкий круг людей по любому из признаков: по возрасту, социальному положению и увлечению.

1.2 Смех и смешное

Смех – характерные прерывистые горловые звуки, вызываемые короткими выыхательными движениями при проявлении веселья, радости, удовольствия и т. п. [2] Смех часто сопровождается чувством радости и удовлетворения. Для каждого человека смешными будут разные вещи, ситуации, фразы и тд. Однако существуют общие критерии и факторы, которые могут быть сигналом того, что что-либо смешное или забавное.

- Неожиданность: Шутка или ситуация, которая нарушает ожидания или преподносит сюрприз;
- Несоответствие: Несоответствие между тем, что мы ожидаем, и тем, что происходит;
- Преувеличение (Гипербола): Утрирование определенных черт или ситуаций;
- Каламбур: Использование слов, которые звучат одинаково, но имеют разное значение;
- Ирония и сарказм: Выражение одного смысла через противоположное утверждение;

1.3 Стили юмора

В своей работе *The Psychology of Humor: An Integrative Approach* [3], в главе «The Humor Styles Questionnaire», Род Мартин представляет четырехфакторную модель стилей юмора. Эта модель, разработанном автором, выделяет четыре основных стиля, различающихся по своей направленности (на себя или на других) и адаптивности.

- **Аффилиативный юмор (Affiliative humor):** Использование юмора для укрепления социальных связей, развлечения окружающих и снятия межличностного напряжения. Этот стиль считается адаптивным и направлен на взаимодействие с другими.
- **Самоподдерживающий юмор (Self-enhancing humor):** Использование юмора для совладания со стрессом, поддержания позитивного взгляда на жизнь и сохранения самоуважения перед лицом невзгод. Этот стиль также адаптивен, но в большей степени ориентирован на внутреннее состояние индивида (внутриличностный).
- **Агрессивный юмор (Aggressive humor):** Использование юмора за счет других людей посредством сарказма, насмешек, критики или унижения. Данный стиль рассматривается как дезадаптивный и межличностный.
- **Самоуничижительный юмор (Self-defeating humor):** Использование чрезмерно самоуничижительного юмора, заискивание перед другими за свой счет, часто в попытке получить одобрение или избежать критики. Этот стиль также считается дезадаптивным и может быть связан как с внутренними проблемами, так и с межличностными отношениями.

Мартин подчеркивает, что первые два стиля способствуют психологическому благополучию, в то время как последние два могут быть связаны с негативными психологическими последствиями.

1.4 Виды шуток

Согласно статье И.В. Вержинской «ЮМОР: ИСТОРИЯ И КЛАССИФИКАЦИЯ ПОНЯТИЙ» [4] виды юмора и шуток делятся на 19 типов:

- 1) **amusement** (развлечение)
- 2) **enjoyment** (наслаждение, удовольствие)
- 3) **fun** (веселье, шутка)
- 4) **pleasure** (удовольствие)
- 5) **absurdity** (нелепость, абсурдность)
- 6) **irony** (ирония)
- 7) **laughableness** (забавность)
- 8) **ludicrousness** (нелепость)
- 9) **ridiculousness** (нелепость)
- 10) **whimsicality** (прихотливость, причуды)

- 11) **wittiness** (остроумие)
- 12) **wryness** (искаженность, противоречивость)
- 13) **burlesque** (бурлеск)
- 14) **caricature** (карикатура)
- 15) **farce** (фарс)
- 16) **jest** (шутка, острота)
- 17) **lampoon** (памфлет, злая сатира)
- 18) **parody** (пародия)
- 19) **satire** (сатира)

Разделение юмора на такое количество видов, используя только техническими решениями почти нереализуемо, вследствии чего будет использоваться классификация по Роду Мартину.

1.5 Сфера применения

Система создания смешных подписей к картинкам может быть применима в следующих сферах:

- создание контента для личных аккаунтов в социальных сетях;
- вирусный маркетинг;
- создание мемов для развлекательных платформ, таких как Reddit, Imgur и др.;
- создание запоминающихся учебных материалов

1.6 Постановка задачи

Необходимо разработать систему, способную распознавать изображение и его смысловое содержание, передаваемое ему пользователем и генерировать смешную фразу, подпись для этого изображения.

Входные данные: изображение.

Выходные данные: смешная подпись, которая может быть использована под данное изображение.

Система должна соответствовать следующим требованиям:

- **высокая скорость обработки:** система должна распознавать изображение и генерировать к нему подпись за время, не превышающее 15 секунд;
- **устойчивость к формату изображения:** система должна сохранять работоспособность при различных форматах, в которых может передоваться изображение, включая: png, jpg, jpeg, bmp, svg.
- **связанность текста:** текст, сгенерированный для картинки должен быть связным, несущим какую-либо информационную нагрузку.

1.7 Обзор существующих аналогов

1.7.1 Supermeme.ai

Supermeme.ai [5] представляет собой онлайн-платформу, разработанную с целью автоматизации процесса создания мемов посредством использования технологий искусственного интеллекта. Основная функция сервиса – генерация мемов на основе текстовых запросов или ключевых слов, при этом система предлагает пользователю шаблоны мемов и автоматически формирует текстовые подписи. Платформа располагает коллекцией популярных мем-шаблонов и предоставляет возможности для модификации сгенерированного текста. Предоставляется функционал для загрузки пользовательских изображений.

1.7.2 Imgflip

Imgflip [6] представляет собой многофункциональную онлайн-платформу, главной задачей которой является предоставление возможностей для создания и обмена визуальным контентом, в том числе мемами. Платформа позволяет добавлять свои подписи к картинкам, а так же генерировать само изображение по промту от пользователя. Проект не предоставляет возможности генерации подписей к картинкам, однако ее можно считать аналогом, так как в конечном итоге пользователь получает картинку со смешной подписью, пусть и написанной самим пользователем.

1.7.3 Pincel

Chat.pincel.app [7] представляет собой онлайн-платформу, предоставляющую доступ к чат-ботам на основе искусственного интеллекта. Основная сфера работы Pincel – создание объемных текстов, однако платформа имеет возможность взаимодействия с изображениями и генерации описания и подписей к ним. Заявляется возможность создания «острых подписей» для фотографий в социальных сетях, так что проект выступает аналогом моей работы.

1.7.4 Magic Hour

Magic Hour AI Meme Generator [8] представляет собой онлайн-инструмент, предназначенный для быстрой генерации мемов с использованием искусственного интеллекта. Пользователь вводит тему или идею, после чего AI генерирует релевантные и юмористические подписи для мемов. Платформа предоставляет библиотеку шаблонов мемов, которые либо автоматически подбираются AI, либо могут быть выбраны самим пользователем. После генерации текст может быть отредактирован.

1.7.5 Funny caption generator

GitHub репозиторий "Funny-caption-generation" [9] от RajeshThevar представляет собой проект, целью которого является генерация смешных подписей к изображениям с использованием методов глубокого обучения. Проект основан на подходе, включающем использование предварительно обученной модели для генерации подписей к изображениям и ее последующей доработки (fine-tuning) на наборе данных, содержащем смешные подписи к изображениям. Основная сфера работы проекта – мультишные изображения. Программа способна самостоятельно выбирать текст надписи на Английском языке и место на изображении, куда эта надпись будет вставлена.

1.7.6 Pallyy

Pallyy Image Caption Generator [10] представляет собой онлайн-инструмент, использующий AI для автоматической генерации подписей к изображениям. Сайт позволяет пользователю загрузить изображение или вписать промт для создания изображения с нуля, выбрать тип подписи: веселая, грустная, умная и т.д. и сгенерировать готовый мем. Среди функционала сайта представлено специальное окно, где пользователь может указать дополнительные пожелания, по генерируемому изображению и тексту.

1.7.7 Aigo.tools

Captionslab на aigo.tools [11] представляет собой бесплатный онлайн-инструмент, использующий искусственный интеллект для генерации подписей к изображениям. Проект работает как веб-сервис, где пользователь вводит ключевые слова или описание изображения в текстовую форму. Затем AI-модель обрабатывает этот ввод и генерирует текстовые варианты подписей. У пользователя нет возможности загрузить изображение самостоятельно, однако данный проект можно считать аналогом моей работы.

1.7.8 Galaxy AI Image Caption Generator

Galaxy AI Image Caption Generator [12] представляет собой онлайн-инструмент, использующий технологии искусственного интеллекта для автоматического создания подписей к загружаемым пользователем изображениям. Основная функция сервиса заключается в анализе загруженной картинки и генерации нескольких вариантов текстовых подписей. Проект предлагает простой интерфейс, где пользователь загружает изображение, после чего AI обрабатывает его и предоставляет сгенерированные подписи.

1.7.9 WritingMate AI Image Caption Generator

WritingMate AI Image Caption Generator [13] представляет собой бесплатный онлайн-инструмент, разработанный на основе искусственного интеллекта для автоматической генерации подписей к изображениям. Пользователи могут загружать изображения различных форматов, после чего AI-алгоритмы WritingMate анализируют визуальное содержание и предлагают разнообразные варианты текстовых подписей.

1.7.10 Gemini

Gemini [14] представляет собой многофункциональную большую языковую модель, разработанную Google AI. В отличие от специализированных инструментов, ориентированных на генерацию подписей к изображениям, Gemini является универсальной нейронной сетью, способной обрабатывать и генерировать различные виды контента, включая текст и изображения. Gemini способен понимать контекст изображения, что позволяет ему генерировать более точные подписи, которые могут быть наполнены разными типами юмора.

1.7.11 Neural Joking Machine

Данное исследование [15] посвящено разработке системы под названием Neural Joking Machine (NJM) для автоматической генерации смешных подписей к изображениям. Проект направлен на создание метода, способного не просто описывать содержимое картинки, но и генерировать подписи, вызывающие смех. В отличие от традиционных подходов к описанию изображений, NJM стремится к моделированию юмора на основе анализа большого количества существующих смешных подписей. Ключевой инновацией является введение метрики "Funny Score" которая используется для количественной оценки и оптимизации "смешности" генерируемых подписей на основе оценочной базы данных. Модель NJM обучалась на созданном авторами датасете BoketeDB, содержащем около миллиона пар "изображение – смешная подпись" собранных с японского веб-сервиса Bokete. Результаты показали, что предложенный метод, использующий предварительно обученную на BoketeDB модель, генерирует подписи, которые по оценкам людей являются более смешными, чем подписи, созданные базовой моделью, хотя и уступают по уровню юмора подписям, написанным людьми. Ключевым результатом является создание фреймворка и метрики, позволяющих количественно подходить к задаче генерации юмористического контента на основе изображений.

1.7.12 Oxford Humorous Image Captions

Данное исследование [16] посвящено представлению OxfordTVG-HIC (Humorous Image Captions) – масштабного датасета для генерации и понимания юмора. Проект направлен на создание ресурса, позволяющего обучать модели для генерации юмористических подписей к

изображениям и изучения механизмов машинного понимания юмора. В отличие от традиционных датасетов для описания изображений, OxfordTVG-HIC характеризуется широким эмоциональным и семантическим разнообразием, включая примеры, способствующие генерации юмора за счет нарушения ожиданий. Датасет содержит около 2.9 миллионов пар "изображение-текст" каждая из которых имеет оценку уровня юмора, что позволяет количественно оценивать "смешность". На основе OxfordTVG-HIC были разработаны модели для автоматической генерации юмористических подписей, а также для оценки уровня юмора сгенерированных текстов.

1.7.13 Humor in AI: Massive Scale Crowd-Sourced Preferences and Benchmarks for Cartoon Captioning

Humor in AI: Massive Scale Crowd-Sourced Preferences and Benchmarks for Cartoon Captioning [17]: Это исследование фокусируется на понимании того, что делает юмор эффективным, и на том, как можно обучать AI-системы генерировать юмористические подписи к изображениям. Авторы создали большой датасет мультишных изображений и краудсорсинговых подписей, где люди оценивали смешность этих подписей. Используя этот датасет, они провели сравнительный анализ различных языковых моделей, чтобы оценить их способность генерировать подписи, которые были бы признаны смешными людьми. Исследование выявило ключевые элементы юмора, такие как игра слов, нарушение ожиданий и культурные отсылки.

1.7.14 AI Humor Generation: Cognitive, Social and Creative Skills for Effective Humor

В этой работе [18] исследуется, может ли AI-генерируемый юмор, использующий когнитивные, социальные и творческие навыки, приблизиться к уровню юмора, создаваемого людьми. В качестве контекста исследования авторы выбрали задачу генерации юмористических подписей к изображениям, публикуемым в социальных сетях. Они разработали метод, включающий наблюдение за изображением, генерацию различных юмористических "углов" и последующую генерацию и ранжирование подписей. Результаты показали, что их метод способен генерировать подписи, которые по уровню юмора сопоставимы с лучшими комментариями пользователей социальных сетей.

1.7.15 XMeCap: Meme Caption Generation with Sub-Image Adaptability

XMeCap: Meme Caption Generation with Sub-Image Adaptability: Это исследование направлено на генерацию подписей именно для мемов, учитывая не только общее содержание изображения, но и детали на его отдельных частях. Авторы подчеркивают важность понимания культурных особенностей и распространенных форматов мемов для создания релевантных и

смешных подписей.

1.7.16 Meme – Generator

GitHub репозиторий "meme-generator" [20] от nikitalokhmachev-ai представляет собой проект, содержащий Python-скрипт для генерации мемов. Скрипт использует библиотеку Pillow для обработки изображений. С технической точки зрения, скрипт принимает на вход путь к изображению, верхний текст и нижний текст, после чего генерирует и сохраняет изображение мема. Данный проект не использует искусственный интеллект для автогенерации подписей, однако его все еще можно рассматривать, как аналог моей работы, так как пользователь в конечном итоге получает мем.

1.7.17 contextual_captions

GitLab репозиторий "contextual_captions" [21] от Sreyasi посвящен задаче генерации контекстуальных подписей к изображениям с использованием глубокого обучения. Проект основан на научной работе и направлен на создание подписей, которые не только описывают содержание изображения, но и отражают его контекст. Реализация проекта выполнена на языке Python с применением TensorFlow или. Архитектура используемой модели представляет собой нейронную сеть, разработанную специально для задачи контекстуального описания изображений. Несмотря на то, что данный репозиторий не фокусируется на создании мемов, его работа направлена на глубокий анализ изображения и его подтекста, что является одной из ключевых задач для генерации смешных подписей.

1.7.18 DeepSeek AI

DeepSeek AI [22] представляет собой компанию, занимающуюся разработкой передовых технологий в области искусственного интеллекта, в частности, больших языковых моделей. В контексте задачи генерации смешных подписей к картинкам, разработки DeepSeek AI представляют интерес благодаря их способности создавать связный и контекстуально релевантный текст. Большие языковые модели, разрабатываемые DeepSeek AI, потенциально могут быть использованы в качестве основы для генерации креативных и юмористических подписей к изображениям.

1.7.19 ChatGPT

ChatGPT [23] представляет собой большую языковую модель, разработанную компанией OpenAI. Она способна генерировать текст, отвечать на вопросы, писать код и выполнять множество других задач, связанных с обработкой естественного языка. В контексте генерации смешных подписей к картинкам, ChatGPT выступает аналогом моей работы благодаря своей способности создавать разнообразный текст и способности к глубокому анализу изображений.

1.7.20 Show and Tell: A Neural Image Caption Generator

Данное исследование [24] представляет собой одну из основополагающих работ в области нейронной генерации подписей к изображениям. В статье авторы предлагают сквозную нейронную сетевую модель, способную автоматически генерировать описания для изображений. Модель, названная "Show and Tell" состоит из сверточной нейронной сети (CNN), используемой в качестве кодировщика для извлечения признаков из изображения, и рекуррентной нейронной сети (RNN), в частности, Long Short-Term Memory (LSTM), в качестве декодировщика для генерации последовательности слов, составляющих подпись. Работа не генерирует смешные подписи к изображениям, однако она вызывает большой интерес, благодаря возможности глубоко анализировать и описывать изображение в контексте его содержания и происходящих на нем событий.

1.7.21 Image Caption Generator

Веб-сайт Image Caption Generator [25] представляет собой бесплатный онлайн-инструмент, использующий искусственный интеллект для автоматической генерации подписей к загружаемым пользователем изображениям. Основная функция сервиса заключается в анализе загруженной картинки и генерации нескольких вариантов текстовых подписей. Пользователю предоставляется возможность выбрать тон подписи (например, смешной) и добавить дополнительный контекст для улучшения релевантности сгенерированного текста. Сайт поддерживает несколько языков.

1.7.22 Junia AI Image Caption Generator

Junia AI Image Caption Generator [26] — это бесплатный инструмент, разработанный для создания контекстуально релевантных подписей к изображениям. Сайт позволяет загрузить изображение или ссылку на это изображение в интернете, добавить дополнительные пояснения и перевести изображение в текст, описывающий его. Проект не выступает полным аналогом моей работы, однако может быть полезен при анализе содержания изображения.

1.7.23 10015.io Image Caption Generator

10015.io Image Caption Generator [27] — это бесплатный онлайн-инструмент, который использует искусственный интеллект для создания подписей к изображениям. Пользователь может загрузить изображение, и инструмент сгенерирует несколько вариантов подписей. Сайт не предлагает никаких дополнительных опций для настройки подписей, однако предлагает на выбор 3 дополнительных подписи, помимо основного описания, сгенерированных специально для постов в социальных сетях.

1.7.24 ZapCap Funny Instagram Caption Generator

ZapCap Funny Instagram Caption Generator [28] — это онлайн-инструмент, предназначенный для генерации смешных подписей для видео в социальных сетях. Пользователь может ввести тему или ключевые слова, связанные с медиафайлом, и инструмент сгенерирует несколько вариантов юмористических подписей. Сайт ориентирован на создание контента для социальных сетей. Проект не способен генерировать подписи к изображениям, однако может принимать в себя описания данного изображения и генерировать подпись уже по нему. Инструмент не выступает полным аналогом моей программы, однако все еще может быть интересен в контексте генерации подписи по описанию в кооперации с другими проектами.

1.7.25 Chat100 Photo Caption Generator

Chat100 Photo Caption Generator [29] — это инструмент, предназначенный для создания подписей к фотографиям. Он позволяет пользователям вводить описание фотографии или ключевые слова, чтобы получить сгенерированные подписи. Пользователь так же может загрузить изображение и описать какую именно подпись он хочет к нему получить.

1.7.26 Simplified

Simplified [30] — это платформа для создания контента, которая предлагает широкий спектр инструментов, включая генератор подписей к изображениям. В контексте моей работы сайт предлагает функционал для создания смешных подписей для изображений в социальных сетях.

1.7.27 Captionslab (aigo.tools)

Captionslab на aigo.tools [11] представляет собой онлайн-инструмент, использующий искусственный интеллект для генерации подписей к изображениям. Пользователь текстово описывает изображение и в каком видео он хочет видеть подпись, однако у пользователя нет возможности загрузить изображение самостоятельно.

1.7.28 YesChat Image Caption Generator

YesChat Image Caption Generator [31] — это инструмент, который использует искусственный интеллект для создания подписей к изображениям. Пользователи могут загружать свои фотографии, и YesChat предоставляет различные варианты подписей.

1.7.29 Thread Creator Image Caption Generator

Thread Creator Image Caption Generator [32] — это инструмент, который использует искусственный интеллект для создания подписей к изображениям. Он позволяет пользователям

загружать изображения и генерировать подписи, выбирая тон и язык. Инструмент ориентирован на создание контента для социальных сетей.

1.7.30 Videotok Caption Generator

Videotok Caption Generator [33] — это онлайн-инструмент, предназначенный для генерации подписей к видео и изображениям. Он позволяет пользователям загружать изображения, вводить ключевые слова или описание контента, чтобы получить сгенерированные подписи.

1.8 Сравнение существующих аналогов

Таблица 1.1 – Сравнительная таблица аналогов

Название	Фокус	AI	Вход	Выход	Лицензия
Supermeme.ai	Генерация мемов	Да	Текст/Ключевые слова, Изображение	Мем	Лицензия сайта
Imgflip	Создание мемов и GIF	Да	Текст, Изображение	Мем, GIF	Лицензия сайта
Pincel	Генерация текста, подписей к изображениям	Да	Текст, Изображение	Описание, подпись	Лицензия сайта
Magic Hour AI	Генерация мемов	Да	Тема/Идея	Мем	Лицензия сайта
Funny caption generator	Генерация смешных подписей	Да	Изображение	Смешная подпись	Лицензия сайта
Pallyy	Генерация подписей	Да	Изображение, Тип подписи, Доп. пожелания	Подпись	Лицензия сайта
Captionslab (aigo.tools)	Генерация подписей	Да	Ключевые слова/Описание	Подпись	Лицензия сайта
Galaxy AI	Генерация подписей	Да	Изображение	Подпись	Лицензия сайта
WritingMate AI	Генерация подписей	Да	Изображение	Подпись	Лицензия сайта

Название	Фокус	AI	Вход	Выход	Лицензия
Gemini	Генерация различного контента	Да	Текст, Изображение	Текст, Изображение	–
Neural Joking Machine	Генерация смешных подписей	Да	Изображение	Смешная подпись	–
Oxford Humorous Image Captions	Генерация и понимание юмора	Да	Изображение	Смешная подпись	–
Humor in AI: Massive Scale Crowd-Sourced Preferences and Benchmarks for Cartoon Captioning	Генерация юмористических подписей к изображениям из мультфильмов	Да	Изображение	Смешная подпись	–
AI Humor Generation: Cognitive, Social and Creative Skills for Effective Humor	Генерация юмористических подписей	Да	Изображение	Смешная подпись	–
XMeCap: Meme Caption Generation with Sub-Image Adaptability	Генерация подписей к мемам	Да	Изображение	Подпись к картинке	–
Meme – Generator	Генерация мемов	Нет	Изображение, Верхний текст, Нижний текст	Мем	MIT License
contxtual cuptions	Генерация контекстуальных подписей	Нет	Изображение и текст	Мем	–
DeepSeek AI	Генерация текста	Да	Фото, текст, описание	Текст	–

Название	Фокус	AI	Вход	Выход	Лицензия
ChatGPT	Генерация текста	Да	Фото, текст, описание	Текст	–
Show and Tell: A Neural Image Caption Generator	Генерация подписей к изображениям	Да	Изображение	Описание картинки	–
Image Caption Generator	Генерация подписей	Да	Изображение и тип подписи	Мем	Лицензия сайта
Junia AI Image Caption Generator	Генерация описания изображения	Да	Изображение или ссылка на него в интернете, дополнительные пояснения	Описание изображения	Лицензия сайта
10015.io Image Caption Generator	Генерация описания и подписей для социальных сетей	Да	Изображение	Описание и подписи для социальных сетей	Лицензия сайта
ZapCap	Генерация смешной подписи по описанию	Да	Описание изображения	Подпись к медиафайлу	Лицензия сайта
Chat100 Photo Caption Generator	Генерация подписей к изображениям	Да	Изображение или описание изображения и стиль опписи	Мем	Лицензия сайта
Simplified	Генерация подписей к изображениям в социальных сетях	Да	Изображение	Смешная подпись	Лицензия сайта
Captionslab	Генерация подписей к изображениям	Да	Текстовое описание	Мем	Лицензия сайта

Название	Фокус	AI	Вход	Выход	Лицензия
YesChat Image Caption Generator	Генерация подписей к изображениям	Да	Текстовое описание / Изображение	Картишка с подписью	Лицензия сайта
Thread Creator Image Caption Generator	Генерация подписей	Да	Изображение	Подпись к изображению	Лицензия сайта
Videotok Caption Generator	Генерация подписей для социальных сетей	Да	Изображение и описание подписи (оpционально)	Подпись для социальных сетей	Лицензия сайта

1.9 Ограничения и этические аспекты практической реализации

При рассмотрении возможности создания конечного продукта на основе исследуемых технологий генерации подписей к изображениям, необходимо учитывать не только технические аспекты, но и существующие ограничения в области контроля и цензурирования генерируемого контента.

На данный момент времени отсутствуют общедоступные и технически совершенные методы для надежного автоматического цензурирования контента, создаваемого в результате взаимодействия модели анализа изображений и генеративной языковой модели (т. е. пары «изображение – подпись»). Существующие подходы, как правило, ограничиваются фильтрацией текста по спискам запрещенных слов или фраз. Такие методы не способны в полной мере:

- учитывать сложный семантический контекст, в котором используется слово или фраза.
- Распознавать иронию, сарказм или другие стилистические приемы, которые могут изменять первоначальный смысл высказывания.
- Анализировать соответствие сгенерированной подписи визуальному содержанию изображения с точки зрения этических норм, культурных особенностей и потенциально оскорбительного или неприемлемого толкования.
- Предотвращать генерацию контента, который, хотя и не содержит явно запрещенных слов, может быть интерпретирован как нарушающий законодательство Российской Федерации (например, разжигание розни, распространение недостоверной информации и т.д.).

В связи с вышеизложенным, несмотря на техническую возможность реализации программного кода, способного генерировать подписи к изображениям, создание и выпуск конеч-

ного продукта для широкого публичного использования сопряжены со значительными рисками. Потенциальная генерация нежелательного, оскорбительного или противоречащего законодательству РФ контента делает практическую реализацию такого продукта невозможной без предварительного решения фундаментальных проблем в области безопасного и этичного ИИ.

Вывод

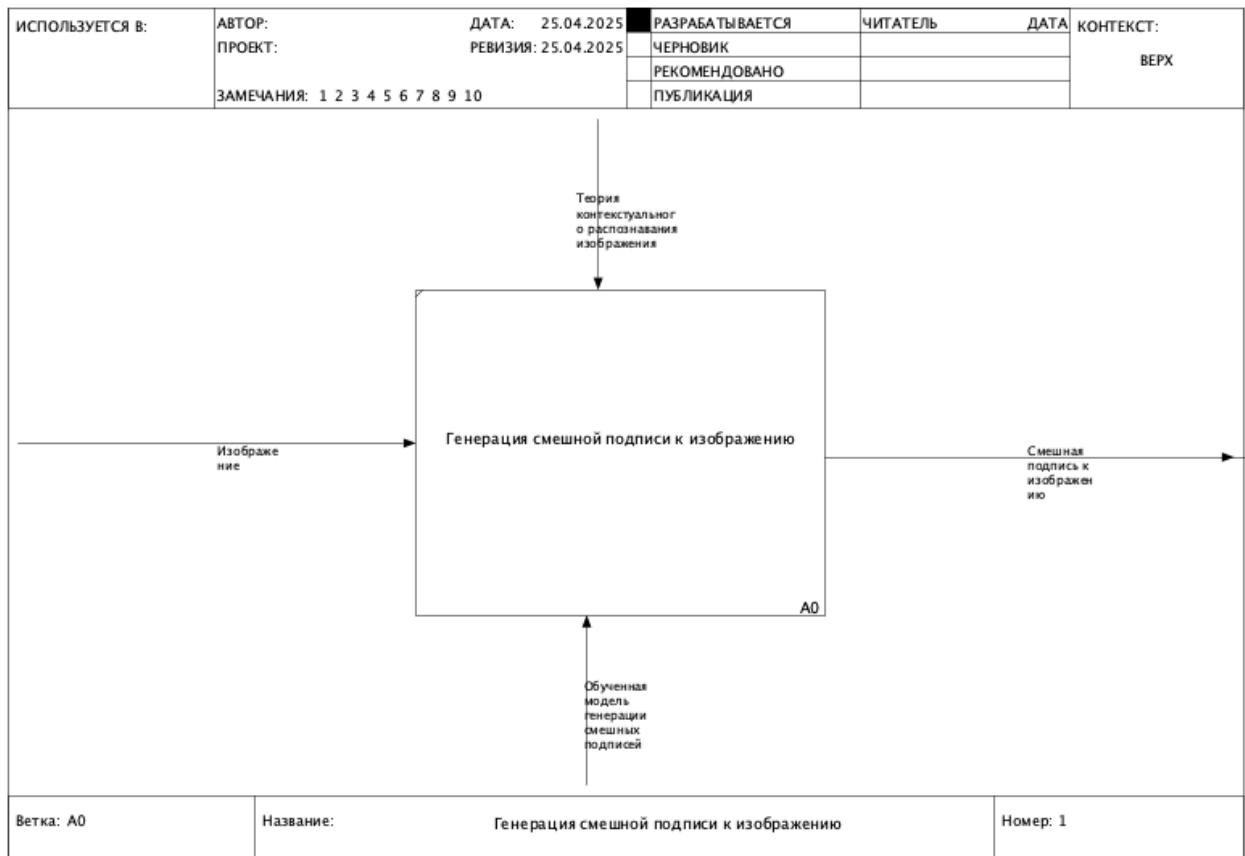


Рисунок 1.1

2 Конструкторская часть

2.1 Общая схема метода

Задача генерации смешной подписи к картинке заключается в создании короткого текста, который в сочетании с визуальным содержанием изображения вызывает у зрителя смех, улыбку или ощущение комичности. На рисунке 2.1 представлена IDEF0-диаграмма первого уровня решаемой задачи.

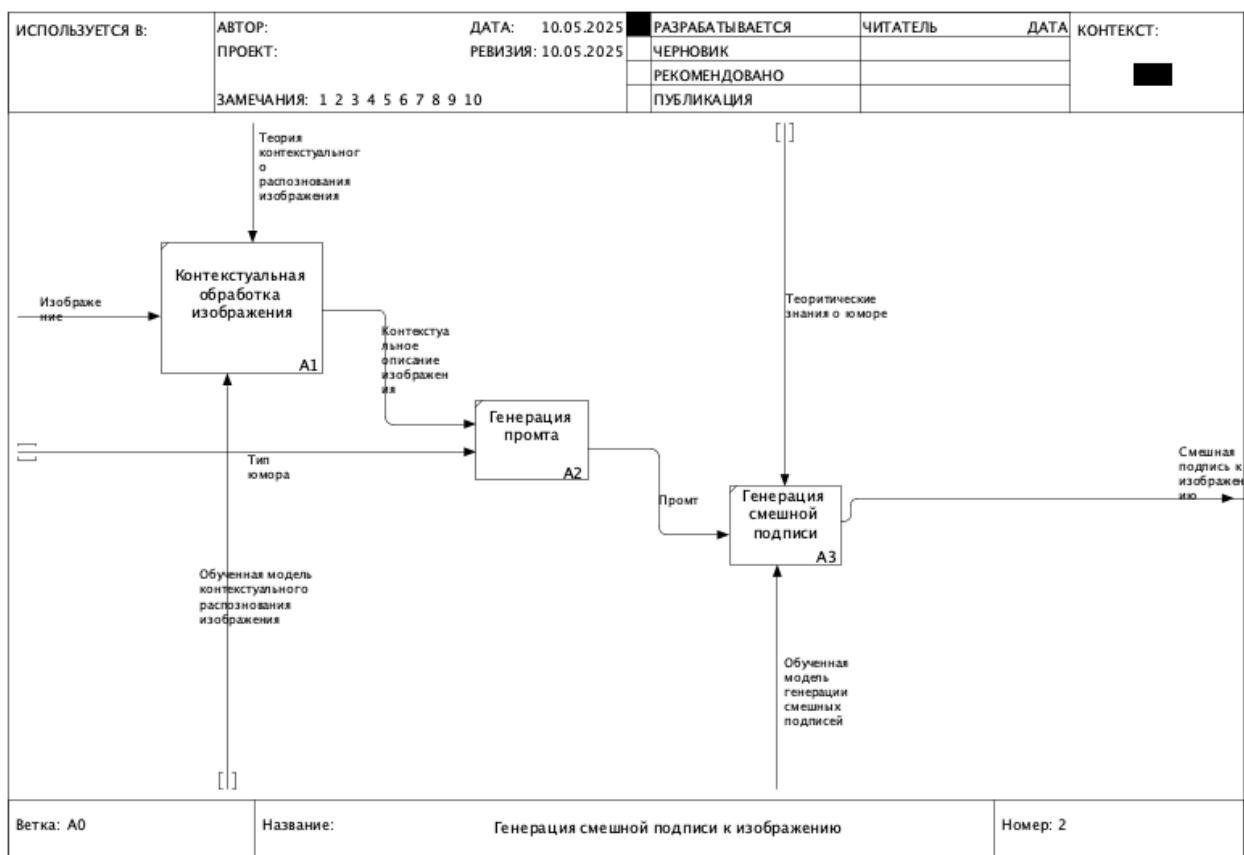


Рисунок 2.1 — IDEF0-диаграмма первого уровня

Перед генерацией подписи мы должны провести анализ изображения и определиться с типом юмора, который пользователь желает увидеть. В первую очередь программа выясняет что изображено на картинке, переданной ей, и какой контекст у этого изображения (блок A1). На основе контекста и содержания изображения программа генерирует контекстуальное описание изображения. Затем у пользователя запрашивается тип юмора который он желает получить и по 2 критериям генерируется запрос (промт) для большой языковой модели (блок A2). Большая языковая модель получает промт, содержащий контекстуальное содержание изображения и информацию о типе юмора и генерирует смешную подпись к данному изображению (блок A3).

2.2 Описание алгоритмов системы

2.2.1 Контекстуальная обработка изображения

Контекстуальная обработка изображения – это подход к анализу изображений, который выходит за рамки изучения отдельных пикселей. Вместо этого учитываются взаимосвязи между объектами, их окружением и общим смыслом сцены. Используя локальный, глобальный и семантический контекст что позволяет алгоритму распознать не только что изображено на картинке, но и ее смысловую нагрузку. Процесс контекстуальной обработки изображения представлен на рисунке 2.2.

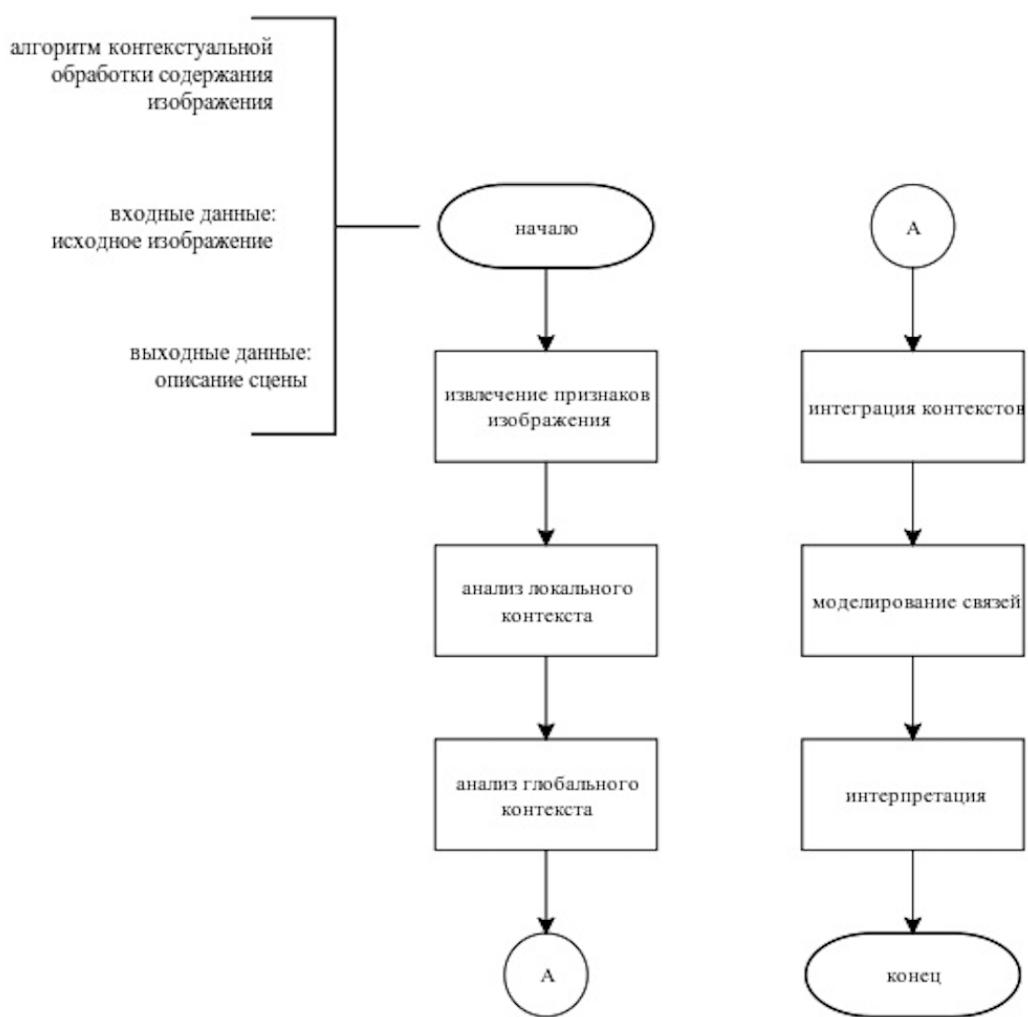


Рисунок 2.2 — Алгоритм работы контекстуальной обработки изображения

Извлечение признаков – это процесс преобразования исходных данных изображения (пикселей) в более информативное и компактное представление. Система выделяет ключевые характеристики (признаки), которые помогают описать содержание изображения. Для корректной работы признаки должны быть различительными, устойчивыми и компактными. Это значит, что признаки должны позволять отличать одни объекты/области от других, не сильно менять-

ся при небольших изменениях освещения, масштаба, ракурса и уменьшать объем данных для обработки.

Анализ локального контекста в обработке изображений заключается в исследовании характеристик пикселя или небольшого региона на основе информации из его непосредственной пиксельной окрестности. Это позволяет выявлять локальные структурные закономерности, такие как границы, текстуры и другие низкоуровневые признаки.

Ключевые методы представлены ниже.

- **Свертка:** Фундаментальная операция, где значение выходного пикселя $O(i, j)$ вычисляется как взвешенная сумма пикселей входного изображения I в окрестности (i, j) с использованием ядра свертки (фильтра) K , где (m, n) пробегают по размерам ядра K . Различные ядра K (например, Гаусса для сглаживания, Собеля для выделения границ) позволяют извлекать разные локальные характеристики.

$$O(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) \cdot K(m, n), \quad (2.1)$$

где

- (i, j) : Координаты текущего пикселя в выходном изображении O .
- (m, n) : Координаты внутри ядра свертки K .
- $I(i-m, j-n)$: Значение пикселя входного изображения I с учетом смещения относительно ядра.

- **Вычисление градиентов:** Оценивает скорость и направление изменения интенсивности пикселей. Часто аппроксимируется с помощью свертки с операторами градиента. Например, для оператора Собеля, компоненты градиента по осям x и y (G_x и G_y) вычисляются сверткой изображения I с соответствующими ядрами K_x и K_y :

$$G_x = I * K_x \quad \text{и} \quad G_y = I * K_y, \quad (2.2)$$

где типичные ядра Собеля:

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.3)$$

Величина градиента G и его направление Θ затем вычисляются как:

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.4)$$

$$\Theta = \text{atan2}(G_y, G_x), \quad (2.5)$$

где

- $\text{atan2}(y, x)$ – арктангенс, возвращающий угол в правильном квадранте.

- **Локальные бинарные шаблоны:** Метод описания текстуры, сравнивающий интенсивность центрального пикселя g_c с интенсивностью P его соседей g_p на окружности радиуса R :

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \quad (2.6)$$

где $s(x) = 1$ если $x \geq 0$ и 0 в противном случае. Это формирует бинарный код, характеризующий локальную текстуру.

Анализ глобального контекста в обработке изображений фокусируется на извлечении и использовании информации, характеризующей изображение в целом или его обширные регионы. Это позволяет системе понимать общую структуру сцены, взаиморасположение крупных объектов и общие статистические свойства изображения.

Ключевые методы представлены ниже.

- **Гистограммные признаки:** Представляют распределение интенсивности пикселей или других локальных признаков по всему изображению. Например, гистограмма интенсивности $h(k)$ для изображения с L уровнями яркости:

$$h(k) = n_k, \quad k = 0, 1, \dots, L - 1, \quad (2.7)$$

где n_k — количество пикселей с интенсивностью k

- **Пространственные пирамидальные представления и Пулинг.**

— **SPM:** Изображение многократно разбивается на все более мелкие сетки (уровни пирамиды). Для каждого региона на каждом уровне вычисляются локальные признаки, которые затем агрегируются.

— **Глобальный усредненный пулинг в CNN:** Слой, который усредняет значения каждой карты признаков на выходе сверточных слоев, преобразуя каждую карту в одно число. Если $F_{c,i,j}$ — активация на c -й карте признаков в позиции (i, j) , то выход GAP для этой карты P_c равен:

$$P_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W F_{c,i,j}, \quad (2.8)$$

где H и W — высота и ширина карты признаков. GAP эффективно агрегирует пространственную информацию в глобальный дескриптор.

- **Механизмы внимания:** Позволяют модели динамически взвешивать важность различных частей входного изображения (или его признаков) при формировании представления. Для каждой части x_i вычисляется вес внимания α_i , и глобальный контекст C может быть представлен как взвешенная сумма всех частей:

$$C = \sum_i \alpha_i x_i \quad (2.9)$$

Веса α_i вычисляются с использованием функции подобия между текущим запросом (query) и ключами (keys), ассоциированными с каждой частью x_i , с последующей нормализацией через softmax:

$$\alpha_i = \text{softmax}(\text{score}(q, k_i)) \quad (2.10)$$

Это позволяет модели избирательно фокусироваться на наиболее релевантной информации по всему изображению.

Семантические признаки – высокоуровневые дескрипторы, которые отражают смысловое содержание изображения или его частей. В отличие от низкоуровневых признаков (границ, текстуры, цвета), которые описывают визуальные характеристики, семантические признаки несут информацию об объектах, их атрибутах, отношениях между ними и общем контексте сцены с точки зрения человеческого понимания.

Интерпретация – завершающий этап конвейера обработки, на котором система, опираясь на признаки, извлеченные с учетом локального, глобального и семантического контекста, формирует окончательный вывод относительно содержания изображения.

— **Классификация на основе признаков:** Для задач классификации (всего изображения или его регионов) используются интегрированные контекстуальные признаки \mathbf{x} . Решение о принадлежности к классу C_k принимается на основе максимизации апостериорной вероятности $P(C_k|\mathbf{x})$. В глубоких нейронных сетях выходные значения (логиты) z_k для каждого класса k преобразуются в вероятности с помощью функции softmax:

$$P(C_k|\mathbf{x}) = \frac{e^{z_k}}{\sum_{j=1}^M e^{z_j}}, \quad (2.11)$$

где M – общее число классов. Принимается решение в пользу класса с максимальной вероятностью: $\hat{C} = \arg \max_k P(C_k|\mathbf{x})$. Контекст, заложенный в \mathbf{x} , помогает разрешать неоднозначности и повышать уверенность в классификации.

— **Оптимизация в вероятностных графических моделях:** Интерпретация заключается в нахождении такой конфигурации меток $\mathbf{s} = (s_1, \dots, s_N)$ для всех пикселей N , которая максимизирует условную вероятность $P(\mathbf{s}|\mathbf{x})$ или, что эквивалентно, минимизирует энергию $E(\mathbf{s}, \mathbf{x})$:

$$E(\mathbf{s}, \mathbf{x}) = \sum_{i=1}^N \psi_U(s_i, \mathbf{x}_i) + \sum_{(i,j) \in \mathcal{N}} \psi_P(s_i, s_j) \quad (2.12)$$

Финальная задача интерпретации заключается в нахождении такой конфигурации меток (\mathbf{s}^*) , которая минимизирует эту общую энергию:

$$\mathbf{s}^* = \arg \min_{\mathbf{s}=(s_1, \dots, s_N)} E(\mathbf{s}, \mathbf{x}) \quad (2.13)$$

Подставляя выражение для энергии, получаем:

$$\mathbf{s}^* = \arg \min_{\mathbf{s}=(s_1, \dots, s_N)} \left(\sum_{i=1}^N \psi_U(s_i, \mathbf{x}_i) + \sum_{(i,j) \in \mathcal{N}} \psi_P(s_i, s_j) \right) \quad (2.14)$$

В этих формулах:

- $\mathbf{s} = (s_1, \dots, s_N)$ – конфигурация меток.
- s_i – метка i -го пикселя.
- (\mathbf{x}_i) – признаки i -го пикселя.
- $(\psi_U(s_i, \mathbf{x}_i))$ – унарный потенциал.
- (\mathcal{N}) – множество пар соседних пикселей.
- $(\psi_P(s_i, s_j))$ – парный потенциал.

2.2.2 Генерация промта

Процесс генерации промта представлен на рисунке 2.3

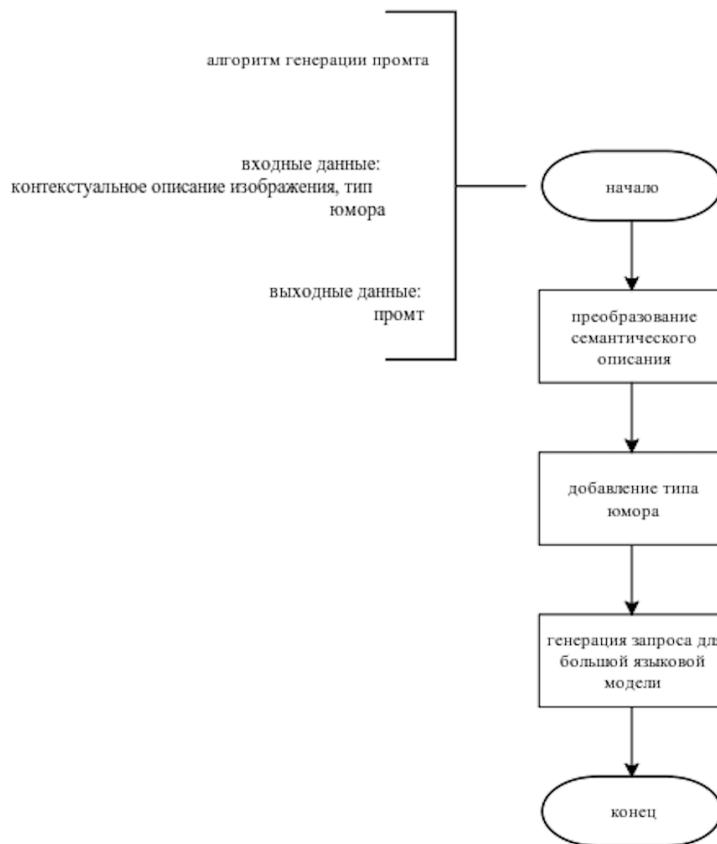


Рисунок 2.3 — Алгоритм работы генератора промтов

Данные из контекстуального обработчика изображений передаются в генератор промта, где они объединяются с типом юмора, указанного пользователем и затем генерируется промт формата: «На изображении ... , придумай смешную подпись, используя ... юмор ». По сути генератор промтов просто приводит все данные к определенной структуре, с которой затем

работает большая языковая модель.

2.2.3 Генерация смешной подписи

Генерация смешной подписи – финальный и ключевой этап в работе программы. По полученному промту и типу юмора большая языковая модель генерирует смешную подпись и предоставляет ее пользователю. Принцип работы генератора подписи показан на рисунке 2.4.

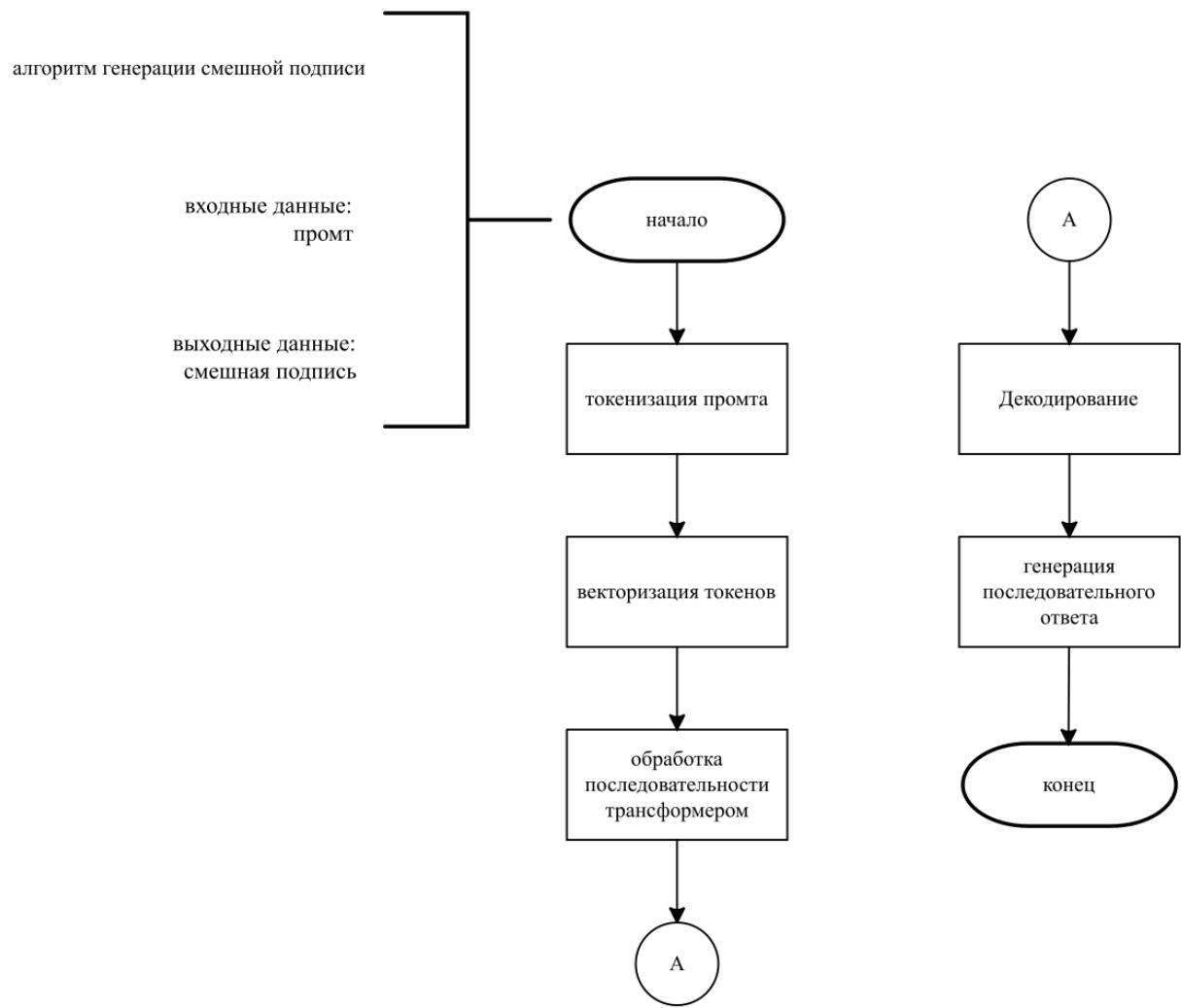


Рисунок 2.4 — Алгоритм работы генератора смешных подписей

Токенизация – это фундаментальный начальный этап обработки текста в моделях обработки естественного языка NLP, включая большие языковые модели LLM. Он заключается в сегментации входной текстовой последовательности S на меньшие единицы, называемые токенами t_i . Эти токены служат атомарными элементами, которые модель может обрабатывать и которым могут быть сопоставлены векторные представления.

В зависимости от типа токенезации токенами могут выступать: слова, части слов и отдельные символы.

Если входной промт S представляет собой строку символов:

$$S = c_1 c_2 \dots c_L, \quad (2.15)$$

где c_j – j -й символ, L – длина строки. После токенизации промт преобразуется в последовательность токенов T :

$$T = (t_1, t_2, \dots, t_N), \quad (2.16)$$

где каждый токен t_i является элементом словаря V , $V = \{v_1, v_2, \dots, v_{|V|}\}$, а N – количество токенов в последовательности (обычно $N \leq L$).

Каждому токену t_i из словаря V сопоставляется уникальный целочисленный идентификатор (ID):

$$ID(t_i) \in \{0, 1, \dots, |V| - 1\} \quad (2.17)$$

Таким образом, токенизованный промт может быть представлен как последовательность целочисленных ID:

$$T_{ID} = (ID(t_1), ID(t_2), \dots, ID(t_N)) \quad (2.18)$$

Векторизация токенов – это процесс преобразования дискретных токенов в плотные числовые векторы фиксированной размерности. Пусть $T_{ID} = (id_1, id_2, \dots, id_N)$ — последовательность целочисленных идентификаторов токенов, полученная после токенизации. Каждому уникальному токену v_j из словаря V (размером $|V|$) сопоставляется вектор эмбеддинга

$$\mathbf{e}_j \in \mathbb{R}^d, \quad (2.19)$$

где d – размерность пространства эмбеддингов. Этот процесс можно представить как функцию отображения E :

$$E \rightarrow \mathbb{R}^d \quad (2.20)$$

или как операцию поиска в матрице эмбеддингов

$$\mathbf{W}_e \in \mathbb{R}^{|V| \times d} \quad (2.21)$$

Каждая строка $\mathbf{W}_{e,j}$ этой матрицы является вектором эмбеддинга для токена с ID j .

Таким образом, последовательность токенов T_{ID} преобразуется в последовательность векторов эмбеддингов

$$\mathbf{X} = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N), \quad (2.22)$$

где $\mathbf{e}_i = \mathbf{W}_{e,id_i}$

Трансформер обрабатывает входную последовательность эмбеддингов $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ через L идентичных слоев. Каждый слой состоит из двух основных подслоев.

— **Многоголовочное Самовнимание:** Каждый токен \mathbf{x}_i формирует Запрос \mathbf{q}_i , Ключ \mathbf{k}_i , Значение \mathbf{v}_i . Выход \mathbf{z}_i для токена i (одна "голова"):

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} = \mathbf{Z} \quad (2.23)$$

где $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ – матрицы запросов, ключей, значений. В Multi-Head Attention h таких операций выполняются параллельно с разными проекциями $\mathbf{W}_j^Q, \mathbf{W}_j^K, \mathbf{W}_j^V$, результаты конкатенируются и проецируются:

$$\text{MultiHead}(\mathbf{X}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)\mathbf{W}^O \quad (2.24)$$

— **Полносвязная сеть прямого распространения FFN:** Применяется к каждому выходу \mathbf{z}_i слоя внимания:

$$\text{FFN}(\mathbf{z}_i) = \max(0, \mathbf{z}_i\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \quad (2.25)$$

Каждый подслой использует **остаточные соединения** ($\mathbf{x} + \text{Sublayer}(\mathbf{x})$) и **нормализацию по слоям** (Layer Normalization). Выход последнего L -го слоя – последовательность контекстуализированных векторов $\mathbf{H} = (\mathbf{h}_1, \dots, \mathbf{h}_N)$. Для генерации используется маскированное самовнимание.

После обработки входного промта Трансформером-энкодером, начинается процесс генерации выходной последовательности $\mathbf{Y} = (y_1, y_2, \dots, y_M)$ токен за токеном.

Основные этапы для генерации токена y_t на шаге t .

1) **Предсказание распределения вероятностей для следующего токена:** На основе контекстуализированного представления промта \mathbf{H} и уже сгенерированных токенов (y_1, \dots, y_{t-1}) , модель вычисляет вектор логитов $\mathbf{l}_t \in \mathbb{R}^{|V|}$ для всех токенов в словаре V . Затем логиты преобразуются в распределение вероятностей $P(y_t | y_{<t}, \mathbf{H})$ с помощью функции softmax:

$$P(y_t = v_j | y_{<t}, \mathbf{H}) = \text{softmax}(\mathbf{l}_t)_j = \frac{\exp(l_{t,j})}{\sum_{k=1}^{|V|} \exp(l_{t,k})} \quad (2.26)$$

где $l_{t,j}$ – логит для j -го токена словаря на шаге t .

2) **Выбор следующего токена y_t :** Из полученного распределения $P(y_t | y_{<t}, \mathbf{H})$ выбира-

ется один токен.

3) **Авторегрессионное продолжение:** Выбранный токен y_t добавляется к последовательности уже сгенерированных токенов, и процесс повторяется для генерации y_{t+1} .

Генерация останавливается при достижении специального токена конца последовательности EOS или максимальной заданной длины.

2.3 Сценарий взаимодействия пользователя с программой

На рисунке 2.5 представлен сценарий взаимодействия пользователя с программой.

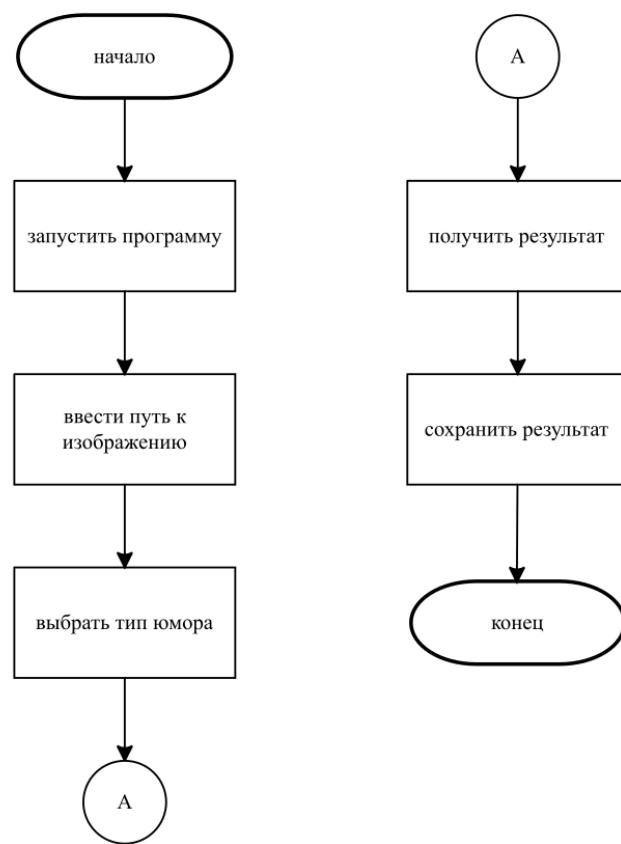


Рисунок 2.5 — Взаимодействие с программой

Вывод

В данной главе представлена конструкторская часть системы генерации смешных подписей к изображениям. Описана общая архитектура метода, основанная на последовательной обработке данных, начиная с анализа визуального контента и заканчивая генерацией текстовой подписи с использованием большой языковой модели.

Разработана и описана общая схема метода в виде IDEF0-диаграммы первого уровня и описаны алгоритмы, лежащие в основе каждого блока системы.

Также представлен сценарий взаимодействия пользователя с программой, иллюстрирующий последовательность действий пользователя и откликов системы.

Таким образом предложенный подход к генерации смешных подписей к картинкам основан на последовательном анализе содержания изображения и последующей генерации подписи с учетом типа юмора, который пользователь желает увидеть.

3 Технологическая часть

3.1 Язык программирования и среда разработки

- В качестве языка программирования для реализации проекта был выбран Python 3.11.8 [34].
- Выбор Python обусловлен следующими факторами:
 - Популярность языка.
 - Наличие большого набора библиотек для ИИ.
 - Пригодность для быстрого прототипирования.
 - Стабильная работа всех используемых библиотек именно на версии 3.11.8.
- Разработка производилась в среде Visual Studio Code [35].

3.2 Выбор средств реализации программного обеспечения

Программное обеспечение состоит из трех модулей.

- Модуль контекстуальной обработки изображения.
- Модуль для автономного перевода текста на русский язык.
- Большая языковая модель для генерации подписей.

При разработке используются следующие библиотеки.

- **Pillow (PIL Fork)** [36]: Библиотека для открытия, обработки и сохранения множества различных форматов изображений. Используется для загрузки и предварительной подготовки изображений перед их передачей в нейронные сети.
- **Transformers** [37]: Библиотека от Hugging Face, предоставляющая state-of-the-art модели для решения широкого круга задач обработки естественного языка (NLP), компьютерного зрения (CV) и мультимодальных задач.
- **PyTorch** [38]: Открытая библиотека машинного обучения, основанная на библиотеке Torch. Является основным фреймворком глубокого обучения, на котором работают модели, используемые через библиотеку Transformers. Обеспечивает тензорные вычисления с поддержкой ускорения на GPU (CUDA).
- **SentencePiece** [39]: Библиотека для токенизации текста, реализующая алгоритмы на основе неконтролируемого обучения для создания словаря из частей слов (subword units).
- **Sacremoses** [40]: Набор инструментов для токенизации и деттокенизации текста.
- **Accelerate** [41]: Библиотека от Hugging Face, упрощающая запуск распределенного обучения и инференса моделей PyTorch на различных аппаратных конфигурациях.
- **BitsAndBytes** [42]: Библиотека, предоставляющая методы 8-битной и 4-битной квантизации для моделей глубокого обучения, что позволяет значительно снизить потребление памяти (RAM и VRAM) при запуске больших моделей с минимальной потерей точности. Используется для запуска LLM на доступном оборудовании.

Кроме этих библиотек, в моей работе используются 3 обученные модели.

- `microsoft/git-large-coco` [43]: Данная модель используется для генерации контекстуального описания входного изображения. Модель GIT (Generative Image-to-text Transformer) представляет собой трансформер, обученный на большом датасете изображений и соответствующих им текстовых описаний (СОКО). Она была выбрана из-за своей способности генерировать достаточно детализированные описания изображений.
- `Helsinki-NLP/opus-mt-en-ru` [44]: Эта модель применяется для осуществления машинного перевода текста с английского языка на русский. Выбор данной модели обусловлен ее высокой эффективностью для пары английский-русский, относительно небольшим размером, что позволяет использовать ее локально без значительных затрат ресурсов.
- `mistral-7b-instruct-v0.2.q4_K_M` [45]: В качестве большой языковой модели (LLM) для генерации непосредственно смешных подписей используется Mistral с 7 миллиардами параметров. Она была выбрана из-за своих достаточно высоких генеративных способностей для создания креативного текста при относительно умеренных (по сравнению с более крупными LLM) требованиях к вычислительным ресурсам, особенно при использовании техник квантизации для локального запуска. Кроме того, эта модель способна работать, используя русский язык.

3.3 Компоненты программного обеспечения

3.3.1 Инициализация моделей

Каждая из моделей требует инициализации. В данном листинге перечислены все глобальные переменные, необходимые для запуска инициализации моделей. Первый блок отвечает за переменные для модели контекстуального изображения, второй блок — модель для перевода текста английского языка на русский, она необходима, так как модель семантического описания изображений использует только английский язык. Последний блок отвечает за переменные для инициализации большой языковой модели, ответственной за генерацию самой подписи.

Листинг 3.1 — Инициализация глобальных переменных для моделей

```
1     img_caption_processor = None
2     img_caption_model = None
3     img_caption_device = "cuda" if torch.cuda.is_available() else "cpu"
4     print(img_caption_device)
5
6     translator_model_name = "Helsinki--NLP/opus-mt-en-ru"
7     translator_tokenizer = None
8     translator_model = None
9     translation_device = img_caption_device
10
11    llm_gguf_model_path = "C:\\\\Users\\\\epifa\\\\Desktop\\\\kursach\\\\code\\\\{"
12        mistral-7b-instruct-v0.2.q4_K_M.gguf}"
13    llm_model_ct = None
14    llm_device = "cuda" if torch.cuda.is_available() else "cpu"
```

3.3.2 Контекстуальное описание изображения

В первую очередь происходит контекстуальное описание изображения. За это отвечает модель «microsoft/git-large-coco» в функции `get_image_caption`. Она на вход получает путь до изображения на устройстве, на выход — контекстуальное описание изображения. Рабочий код представлен в листинге 3.3.

Листинг 3.2 — Получение семантического описания

```
1  def get_image_caption(image_path_or_url):
2      global img_caption_processor, img_caption_model
3          if img_caption_model is None:
4              initialize_image_caption_model()
5          if img_caption_model == "ERROR":
6              return f"ERROR", None
7
8      try:
9          if image_path_or_url.startswith("http"):
10              raw_image = Image.open(requests.get(image_path_or_url,
11                                              stream= True).raw).convert('RGB')
12          else:
13              raw_image = Image.open(image_path_or_url).convert('RGB')
14          inputs = img_caption_processor(images=raw_image,
15                                          return_tensors="pt").to(img_caption_device)
16          out = img_caption_model.generate(**inputs, max_length
17                                           =200, num_beams =15, early_stopping=True)
18          english_caption = img_caption_processor.decode(out[0],
19                                                       skip_special_tokens=True)
20
21      return english_caption, None
```

3.3.3 Перевод описания на русский

Модель для описания изображения способна функционировать только на английском языке, в то время как генерация подписи происходит полностью на русском языке, поэтому описание должно быть переведено на русский язык перед дальнейшим использованием. Это происходит с использованием обученной модели «Helsinki–NLP/opus-mt-en-ru» в функции `translate_text_local`, которая получает на вход текст на английском и возвращает этот текст, переведенный на русский язык. Код функции представлен в листинге 3.3.

Листинг 3.3 — Перевод подписи на русский

```
1 def translate_text_en_to_ru(text_to_translate):
2     global translator_en_ru_tokenizer, translator_en_ru_model
3     if translator_en_ru_model == "ERROR" or not text_to_translate or not
4         isinstance(text_to_translate, str):
5         return f"ERROR_TRANSLATE_EN_RU: {text_to_translate if isinstance(
6             text_to_translate, str) else ''}"
7     try:
8         inputs = translator_en_ru_tokenizer(text_to_translate,
9             return_tensors="pt", padding=True, truncation=True, max_length
10            =512).to(translation_device)
11         translated_ids = translator_en_ru_model.generate(**inputs,
12             max_length=512, num_beams=5, early_stopping=True)
13         raw_translated_text = translator_en_ru_tokenizer.decode(
14             translated_ids[0], skip_special_tokens=True)
15         if check_text_for_banwords(raw_translated_text):
16             return "BAN_CONTENT_RUS_DESC"
17         return raw_translated_text
18     except Exception as e:
19         return f"ERROR_TRANSLATE_EN_RU: [{e}] {text_to_translate}"
```

3.3.4 Генерация смешной подписи

После получения семантического описания изображения на русском языке, скрипт переходит к генерации смешной подписи с использованием модели «mistral-7b-instruct-v0.2.Q4_K_M». Это происходит в функции `generate_funny_caption_llm`, она представлена в листинге 3.5. На вход поступает описание изображения и тип юмора, выбранный пользователем. На выходе получаем смешную подпись. Функция формирует запрос для большой языковой модели в формате: `system_prompt = ("Ты — остроумный ассистент, который создает короткие и смешные подписи к описаниям изображений на русском языке. Твоя задача — придумать только одну смешную подпись. Не пиши ничего лишнего, только саму подпись. Строго придерживайся заданного стиля юмора.")` Промт отправляется большой языковой модели и в ней формируется ответ (листинг 3.4).

Листинг 3.4 — Получение токенизированного ответа

```
1 try:
2     generated_caption = llm_model_ct(
3         prompt_for_mistral,
4         max_new_tokens=70,
5         temperature=0.7,
6         top_p=0.9,
7         top_k=40,
8         repetition_penalty=1.15,
9         stop=["\n\n", "[INST]", "</s>"],
10        stream=False
11    )
12    funny_caption_rus = generated_caption.strip()
13    end_time = time.time()
14    if "\n" in funny_caption_rus:
15        lines = [line.strip() for line in funny_caption_rus.split('\n')]
16        if line.strip():
17            funny_caption_rus = lines[0] if lines else ""
18    prefixes_to_remove = []
19    funny_caption_rus_lower = funny_caption_rus.lower()
20    for prefix in prefixes_to_remove:
21        if funny_caption_rus_lower.startswith(prefix):
22            funny_caption_rus = funny_caption_rus[len(prefix):].strip()
23    if not funny_caption_rus or len(funny_caption_rus) < 3:
24        return "LLM_EMPTY_OR_SHORT_RUS"
25
26    if check_text_for_banwords(funny_caption_rus):
27        return "BAN_CONTENT_LLM_OUTPUT_RUS"
28    return funny_caption_rus
```

3.3.5 Взаимодействие с пользователем

Взаимодействие с пользователем происходит путем общения через консольный интерфейс программы. После запуска программы, она запрашивает у пользователя.

- 1) Выбрать файл изображения
- 2) Выбрать тип юмора

После чего генерируется подпись и она возвращается пользователю.

3.3.6 Установка и запуск

Для установки потребуется язык программирования Python. Сначала необходимо загрузить пакеты, используемые программой. Для этого следует ввести команду:

```
pip install -r requirements.txt
```

После этого для запуска используется команда:

```
python main.py
```

После запуска необходимо следовать инструкции, расположенной в терминале.

3.3.7 Физические компоненты системы

Система генерации смешных подписей к изображениям размещена на одном устройстве.

Диаграмма развертывания представлена на рисунке 3.1.

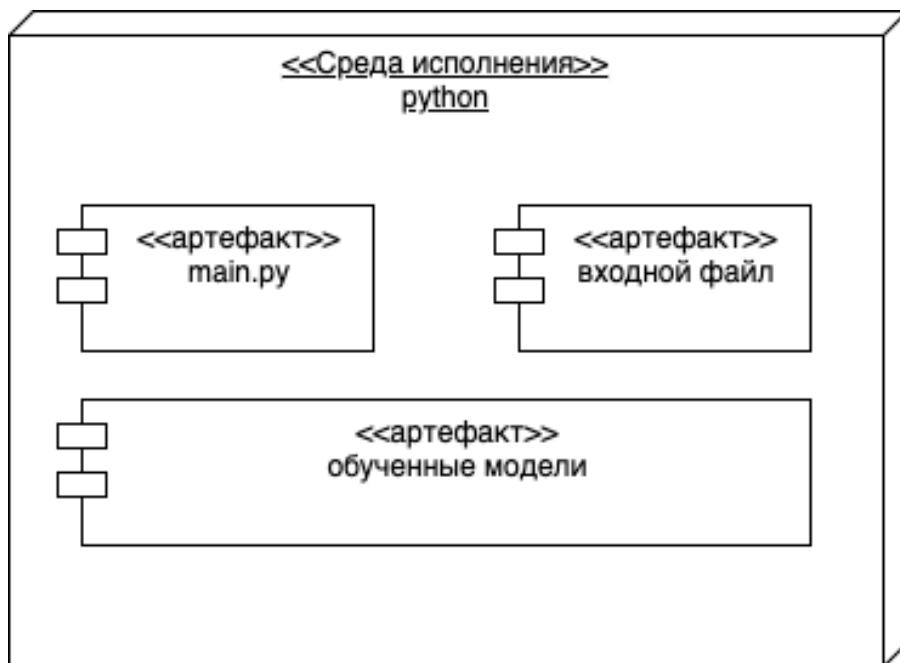


Рисунок 3.1 — Диаграмма развертывания

Вывод

В технологической части представлена программная реализация системы для генерации смешных подписей к изображениям, реализованной на Python с использованием больших языковых моделей. Решение состоит из трех основных модулей: семантическое описание изображения, генерация смешной подписи к изображению и перевод этой подписи на русский. Для повышения производительности реализована обработка текста и изображения с использованием видеокарты.

4 Исследовательская часть

Целью данной исследовательской работы является изучение зависимости времени выполнения и потребления вычислительных ресурсов (в частности, видеопамяти) от характеристик входного изображения при генерации юмористических подписей с использованием двух различных больших языковых моделей.

4.1 Характеристики тестового стенда

Исследование проводилось на персональном компьютере (ЭВМ) со следующими техническими характеристиками:

- операционная система: Windows 10
- Оперативная память (ОЗУ): 16 ГБ DDR4
- Центральный процессор (ЦП): Intel Core i7-7700K;
- Графический процессор (ГП): NVIDIA GeForce RTX 3060 с 12 ГБ видеопамяти (VRAM) GDDR6.

4.2 Методика оценки производительности и потребления ресурсов

Все тесты выполнялись с использованием технологии NVIDIA CUDA, что подразумевает проведение вычислений преимущественно на графическом процессоре (GPU) и активное использование его видеопамяти (VRAM). В исследовании сравнивались две большие языковые модели (LLM):

- 1) `mistral-7b-instruct-v0.2.Q4_K_M.gguf` (далее – Mistral 7B GGUF) – квантованная версия 7-миллиардной модели Mistral, оптимизированная для локального запуска.
- 2) `meta-llama/Meta-Llama-3-8B-Instruct` (далее – Llama 3 8B Instruct) – 8-миллиардная инструктивная модель Llama 3, загружаемая через библиотеку Transformers.

Для каждого из 10 различных тестовых изображений было проведено по 8 независимых запусков для каждой LLM с целью сбора данных о времени генерации подписи и пиковом потреблении VRAM. Таким образом, на каждое изображение приходилось 16 измерений (по 8 для Mistral 7B GGUF и Llama 3 8B Instruct). В рамках этих 8 запусков для каждой модели варьировались типы запрашиваемого юмора (по 2 запуска на каждый из 4-х типов юмора, определенных в программной реализации).

Время инициализации (загрузки в память) самих моделей не включалось в замеры, чтобы оценить непосредственно время, затрачиваемое на обработку изображения и генерацию текстовой подписи. Пиковое потребление видеопамяти отслеживалось с помощью стандартного системного инструмента «Диспетчер задач» Windows во время выполнения этапа генерации.

Усредненные по 8 запускам результаты для каждого тестового изображения и каждой LLM сведены в таблицу 4.1.

Таблица 4.1 — Сравнение обученных моделей

Номер изображения	Mistral Время(Сек)	Mistral VRAM(ГБ)	Llama Время(Сек)	Llama VRAM(ГБ)
1	13.74	11.3	12.56	12
2	12.52	11.7	11.37	12
3	11.43	11.1	13.21	12
4	14.43	12	15.32	12
5	13.61	11.6	11.13	11.8
6	12.57	11.5	12.39	12
7	11.97	11.2	10.37	12
8	11.56	11.7	12.29	12
9	12.51	11.8	13.22	12
10	12.30	11.5	12.11	12

Модель Llama 3 8B Instruct, стремясь к максимальному использованию доступной видеопамяти. На тестовом стенде с 12 ГБ VRAM она практически во всех случаях (за исключением одного измерения для изображения №5, где зафиксировано 11.8 ГБ) потребляла все 12 ГБ. Это указывает на то, что даже при использовании 4-битной квантизации, данная модель является достаточно требовательной к объему VRAM.

Модель Mistral 7B GGUF (7 миллиардов параметров, квантизация Q4_K_M) показала умеренное потребление видеопамяти, в среднем около 11.54 ГБ (диапазон от 11.1 ГБ до 12.0 ГБ). Для изображения №4 потребление VRAM этой моделью достигло максимальных 12.0 ГБ, сравнявшись с Llama 3. В остальных случаях Mistral 7B GGUF оставляла небольшой резерв VRAM.

Касательно времени выполнения, среднее время генерации для Mistral 7B GGUF составило приблизительно 12.66 секунд, в то время как для Llama 3 8B Instruct – около 12.40 секунд. Llama 3 в среднем оказалась на 2% быстрее, чем Mistral 7B GGUF в данных исследованиях.

4.3 Данные по тестируемым изображениям

Ниже представлены тестовые изображения, использованные в ходе исследования, с указанием усредненных значений времени генерации подписи и пикового потребления видеопамяти для каждой из исследуемых языковых моделей. Примеры работы программы не могут быть предоставлены в связи с отсутствием жестких методов цензуры пар «Изображение – подпись». Генерируем контент может противоречить законам РФ.



Изображение 1. Mistral: 13.74 с, 11.3 ГБ VRAM. Llama: 12.56 с, 12.0 ГБ VRAM.



Изображение 2. Mistral: 12.52 с, 11.7 ГБ VRAM. Llama: 11.37 с, 12.0 ГБ VRAM.



Изображение 3. Mistral: 11.43 с, 11.1 ГБ VRAM. Llama: 13.21 с, 12.0 ГБ VRAM.



Изображение 4. Mistral: 14.43 с, 12.0 ГБ VRAM. Llama: 15.32 с, 12.0 ГБ VRAM.



Изображение 5. Mistral: 13.61 с, 11.6 ГБ VRAM. Llama: 11.13 с, 11.8 ГБ VRAM.



Изображение 6. Mistral: 12.57 с, 11.5 ГБ VRAM. Llama: 12.39 с, 12.0 ГБ VRAM.



Изображение 7. Mistral: 11.97 с, 11.2 ГБ VRAM. Llama: 10.37 с, 12.0 ГБ VRAM.



Изображение 8. Mistral: 11.56 с, 11.7 ГБ VRAM. Llama: 12.29 с, 12.0 ГБ VRAM.



Изображение 9. Mistral: 12.51 с, 11.8 ГБ VRAM. Llama: 13.22 с, 12.0 ГБ VRAM.



Изображение 10. Mistral: 12.30 с, 11.5 ГБ VRAM. Llama: 12.11 с, 12.0 ГБ VRAM.

Вывод

В данном разделе проведено исследование работы программы с двумя различными LLM на 1 компьютере для 10 изображений. Представлены данные по времени работы программы и затратам VRAM. В ходе исследования было установлено, что обе LLM справляются с поставленной задачей, работают от 11 до 15 секунд и потребляют от 11 до 12Гб VRAM. Результаты работы программы не могут быть представлены в связи с потенциальными нарушениями законов Российской Федерации.

5 Организационно-правовая часть

Разрабатываемое программное обеспечение (далее – **ПО**) предназначено для генерации юмористических подписей к изображениям, загружаемым пользователем. В основе **ПО** лежат технологии искусственного интеллекта, в частности, две нейронные сети: первая осуществляет семантический анализ предоставленного изображения, вторая – генерирует текстовую подпись на основе результатов анализа и выбранного пользователем стиля юмора. Использование и распространение **ПО** и результатов его работы регулируется действующим законодательством Российской Федерации.

5.1 Нейронные сети и искусственный интеллект в контексте российского законодательства

На текущий момент в российском законодательстве отсутствует исчерпывающее юридическое определение термина «нейронная сеть». Однако, технологии, основанные на нейронных сетях, являются ключевым компонентом систем искусственного интеллекта (ИИ). Развитие ИИ регулируется рядом стратегических документов, таких как Национальная стратегия развития искусственного интеллекта на период до 2030 года [46]. С точки зрения правового статуса, программное обеспечение, реализующее нейронные сети, рассматривается как программа для ЭВМ и объект авторского права, подпадающий под действие Части четвертой Гражданского кодекса Российской Федерации [47].

Важно отметить, что использование технологий ИИ, включая нейронные сети, для создания контента накладывает определенные обязательства по соблюдению законодательства в сфере информации, защиты данных и противодействия распространению запрещенной информации.

5.2 Проблемы цензурирования и ограничения практического использования ПО

Одной из ключевых проблем, препятствующих безопасному практическому использованию разрабатываемого **ПО**, является отсутствие на сегодняшний день эффективных и технически реализуемых методов «жесткого» (гарантированного) цензурирования генерируемых пар «изображение – текстовая подпись».

Хотя в **ПО** могут быть интегрированы механизмы фильтрации на основе списков запрещенных слов и выражений (так называемые «бан-листы»), такие подходы имеют существенные ограничения.

— **Контекстуальная слепота:** Фильтры по ключевым словам не способны анализировать контекст, в котором используется слово, что может приводить как к ложным срабатываниям, так и к пропуску нежелательного контента, выраженного иносказатель-

но.

- **Неспособность оценить семантическое соответствие и этичность:** Автоматические фильтры не могут в полной мере оценить, насколько сгенерированная подпись уместна, этична и не оскорбительна в привязке к конкретному визуальному содержанию изображения.
- **Вероятностная природа генерации:** Нейронные сети, используемые для генерации текста, по своей природе являются вероятностными моделями. Это означает, что даже при наличии фильтров всегда существует ненулевая вероятность генерации непредвиденного, нежелательного или противоправного контента.

Комплексная оценка вывода **ПО** и его соответствия законодательству и этическим нормам требует экспертного человеческого анализа каждой сгенерированной пары «изображение – подпись». Автоматизация такого процесса на текущем уровне развития технологий невозможна с достаточной степенью надежности.

В связи с вышеизложенным, несмотря на то, что программный код, реализующий алгоритмы анализа изображений и генерации подписей, может быть технически создан и отложен, реализация конечного продукта, предназначенного для широкого или неконтролируемого использования, не представляется возможной. Отсутствие гарантированных методов предотвращения генерации текста, потенциально противоречащего правилам и законам Российской Федерации, делает такое использование недопустимым.

Это ограничение также влияет на выбор возрастного рейтинга. Даже если стремиться к созданию контента общего характера, невозможность полной фильтрации нежелательных тем и лексики обуславливает необходимость присвоения ПО возрастного ограничения «18+» в соответствии с Федеральным законом от 29.12.2010 № 436-ФЗ «О защите детей от информации, причиняющей вред их здоровью и развитию».

5.3 Лицензионная часть

Таблица 5.1 — Лицензии используемых программных продуктов и моделей

Программный продукт/Модель	Лицензия
Python	PSF License
PyTorch	BSD-3-Clause License
requests	Apache 2.0 License
transformers (Hugging Face)	Apache 2.0 License
ctransformers	MIT License
cryptography	Apache 2.0 / BSD License
Helsinki-NLP/opus-mt-en-ru	Apache 2.0 License
microsoft/git-large-coco	MIT License
TheBloke/Mistral-7B-Instruct-v0.2-GGUF (модель)	Apache 2.0 License

Таблица 5.2 — Сравнительные характеристики основных используемых лицензий

Критерий	Apache 2.0	MIT	BSD-3-Clause	PSF
Коммерческое использование	Да	Да	Да	Да
Модификация	Да	Да	Да	Да
Распространение	Да	Да	Да	Да
Ограничение ответственности	Да	Да	Да	Да
Сохранение уведомления об авторских правах и тексте лицензии	Да	Да	Да	Да
Указание изменений	Да	Нет	Нет	Да (в документации)
Предоставление исходного кода (при распространении бинарников)	Нет	Нет	Нет	Нет
Предоставление патентных прав	Да (явное)	Неявное	Неявное	Да (явное)
Использование товарных знаков	Нет	Нет	Нет	Нет
Сублицензирование	Да	Да	Да	Да
Копилефт	Нет	Нет	Нет	Нет
Запрет на использование имени для продвижения (No Endorsement Clause)	Ограничено	Нет	Да (явный)	Ограничено
Совместимость с GPL	Да (с GPLv3+)	Да (с GPLv2+)	Да (с GPLv2+)	Да (с GPLv2+)

5.4 Выбор лицензии для разрабатываемого ПО

Анализ лицензионных условий используемых компонентов (см. Таблицу 5.1) показал, что проект включает библиотеки с различными открытыми лицензиями, такими как Apache License 2.0, BSD 3-Clause и Python Software Foundation License (PSF).

Все используемые библиотеки имеют в целом совместимые разрешительные лицензии, которые допускают коммерческое использование, модификацию, распространение и включение в состав более крупных проектов.

Однако, необходимо учитывать, что PSF License, под которой распространяется язык программирования Python, используемый в проекте, имеет специфические условия, касающиеся использования имени "Python" и связанных с ним товарных знаков, а также требования к документации при распространении модифицированных версий или продуктов, построенных с использованием Python. Эти условия не полностью покрываются стандартной лицензией Apache 2.0.

С учётом лицензионной совместимости большинства используемых компонентов (под Apache-2.0, BSD, MIT, которые могут быть свободно интегрированы в Apache-2.0 проект) и специфических требований PSF, для кода самого разрабатываемого программного продукта выбирается Apache License 2.0 с ограничениями PSF. Этот выбор обусловлен ее разрешительным характером, явным предоставлением патентных прав и требованиями к прозрачности изменений.

Такой подход обеспечит возможность коммерческого распространения, модификации, включит положения о патентной защите для вашего кода и при этом будет полностью соответствовать особым условиям PSF по защите торговой марки Python и соблюдению требований к документации при распространении. Полный текст лицензии Apache 2.0 с ограничениями PSF прилагается в приложении А.

Вывод

В рамках данной главы были рассмотрены организационно-правовые аспекты, связанные с разработкой и потенциальным использованием программного обеспечения для генерации юмористических подписей к изображениям.

Установлено, что хотя само ПО, основанное на технологиях нейронных сетей, подпадает под общие нормы регулирования программ для ЭВМ и технологий искусственного интеллекта, ключевым препятствием для его практического внедрения и использования в качестве конечного продукта является отсутствие надежных методов цензурирования генерируемых пар «изображение – текст». Вероятностный характер нейросетевой генерации и неспособность существующих фильтров осуществлять глубокий семантический и этический анализ делают невозможным гарантированное предотвращение создания контента, противоречащего законодательству Российской Федерации и общепринятым нормам.

Таким образом, несмотря на техническую возможность создания программного кода, реализующего заявленную функциональность, разработанное ПО не подлежит использованию в качестве публичного сервиса или продукта без предварительной экспертной оценки каждого случая генерации контента. Это ограничение является фундаментальным на текущем этапе развития технологий контроля ИИ-генерируемого контента.

Для кода самого разрабатываемого ПО выбрана пермиссивная лицензия Apache 2.0. Она предоставляет широкие права на использование, модификацию и распространение кода, включая коммерческие цели, при этом требуя сохранения уведомлений об авторских правах, указания внесенных изменений и предоставляя явные патентные права. При этом, учитывая использование языка Python, при распространении продукта необходимо обеспечить полное соблюдение специфических условий PSF License, касающихся использования товарного знака Python и требований к документации. Такой подход обеспечивает гибкость использования собственного кода в исследовательских и образовательных целях под Apache 2.0, при этом явно указывая на отсутствие гарантий со стороны разработчика и перекладывая ответственность за использование результатов работы ПО на конечного пользователя (с учетом необходимости соблюдения всех применимых лицензий), который должен быть осведомлен о рисках, описанных выше. Вопрос возрастной маркировки «18+» сохраняет актуальность даже для исследовательского прототипа ввиду невозможности полного контроля генерируемого контента.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была достигнута поставленная цель – разработано программное решение, использующее методы машинного обучения для генерации смешных подписей к изображениям.

В процессе работы были решены следующие задачи: проведён обзор существующих аналогов и подходов, выбран метод на основе предобученной модели, спроектировано и реализовано программное решение, а так же определен правовой статус полученного решения.

На основе тестирования была получена информация о ресурсах потребляемых программой для своей работы. В среднем обработка одного изображения занимает 12.64 секунды, при этом потребляется 11.7 Гб VRAM.

Полученные результаты подтверждают работоспособность предложенного подхода и соответствие программного решения поставленным задачам.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Kartaslov [Электронный ресурс]. – Режим доступа: <https://kartaslov.ru/>
2. Kartaslov [Электронный ресурс]. – Режим доступа: <https://kartaslov.ru/>
3. Martin, R. A. (2007). *The Psychology of Humor: An Integrative Approach*. Academic Press. [c. 232-236]
4. Вержинская, И. В. (2011). ЮМОР: ИСТОРИЯ И КЛАССИФИКАЦИЯ ПОНЯТИЙ.
5. Supermeme.ai [Электронный ресурс]. – Режим доступа: <https://supermeme.ai/>.
6. Imgflip [Электронный ресурс]. – Режим доступа: <https://imgflip.com/memegenerator>
7. Chat.pincel.app [Электронный ресурс]. – Режим доступа: <https://chat.pincel.app/login>
8. Magic Hour AI Meme Generator [Электронный ресурс]. – Режим доступа: <https://magichour.ai/products/ai-meme-generator>
9. Funny-caption-generation [Электронный ресурс]. – Режим доступа: <https://github.com/RajeshThevar/Funny-caption-generation>
10. Pallyy Image Caption Generator [Электронный ресурс]. – Режим доступа: <https://pallyy.com/tools/image-caption-generator>
11. Captionslab – AI Caption Generator [Электронный ресурс]. – Режим доступа: <https://aigo.tools/best/captionslab>
12. Galaxy AI Image Caption Generator [Электронный ресурс]. – Режим доступа: <https://galaxy.ai/ai-image-caption-generator>
13. WritingMate AI Image Caption Generator [Электронный ресурс]. – Режим доступа: <https://writingmate.ai/free-tools/image-caption-generator>
14. Gemini [Электронный ресурс]. – Режим доступа: <https://gemini.google.com/app>.
15. Uchida, Y., Akita, K., & Aizawa, A. (2018). Neural Joking Machine: Humorous image captioning.
16. Li, R., Sun, S., Elhoseiny, M., & Torr, P. (2023). OxfordTVG-HIC: Can Machine Make Humorous Captions from Images?

17. Zhang, J., Jain, L., Guo, Y., Chen, J., Zhou, K. L., Suresh, S., ... & Wagenmaker, A. (2024). Humor in AI: Massive Scale Crowd-Sourced Preferences and Benchmarks for Cartoon Captioning.
18. Kim, S., & Chilton, L. B. (2024). AI Humor Generation: Cognitive, Social and Creative Skills for Effective Humor.
19. Chen, Y., Yan, S., Zhu, Z., Li, Z., & Xiao, Y. (2024). XMeCap: Meme Caption Generation with Sub-Image Adaptability.
20. nikitalokhmachev-ai. meme-generator. [Электронный ресурс]. – Режим доступа: <https://github.com/nikitalokhmachev-ai/meme-generator>
21. Sreyasi. contextual_captions. [Электронный ресурс]. – Режим доступа: https://git.jl-k.com/Sreyasi/contextual_captions.
22. DeepSeek AI. [Электронный ресурс]. – Режим доступа: <https://deepseek.ai/>.
23. ChatGPT – OpenAI. [Электронный ресурс]. – Режим доступа: <https://openai.com/chatgpt>.
24. Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator.
25. Image Caption Generator. [Электронный ресурс]. – Режим доступа: <https://imagecaptiongenerator.com>
26. Junia AI Image Caption Generator. [Электронный ресурс]. – Режим доступа: <https://www.junia.ai/tools/image-caption-generator>
27. 10015.io Image Caption Generator. [Электронный ресурс]. – Режим доступа: <https://10015.io/tools/image-caption-generator?ysclid=m9tvuue27g431228099>
28. ZapCap Funny Instagram Caption Generator. [Электронный ресурс]. – Режим доступа: <https://zapcap.ai/tools/funny-instagram-caption-generator/>
29. Chat100 Photo Caption Generator. [Электронный ресурс]. – Режим доступа: <https://chat100.ai/features/photo-caption-generator>
30. Simplified. [Электронный ресурс]. – Режим доступа: <https://app.simplified.com/>
31. YesChat Image Caption. [Электронный ресурс]. – Режим доступа: <https://www.yeschat.ai/features/image-caption-generator>
32. Thread Creator Image Caption Generator. [Электронный ресурс]. – Режим доступа: <https://threadcreator.com/tools/image-caption-generator>

33. Videotok Caption Generator. [Электронный ресурс]. – Режим доступа: <https://www.videotok.app/caption-generator>
34. Python 3.11.8. [Электронный ресурс]. – Режим доступа: <https://www.python.org/downloads/release/python-3118/>
35. Visual Studio Code. [Электронный ресурс]. – Режим доступа: <https://code.visualstudio.com/>
36. Pillow (PIL Fork) Documentation. [Электронный ресурс]. – Режим доступа: <https://pillow.readthedocs.io/en/stable/>
37. Hugging Face Transformers. [Электронный ресурс]. – Режим доступа: <https://huggingface.co/docs/transformers/index>
38. PyTorch. [Электронный ресурс]. – Режим доступа: <https://pytorch.org/>
39. SentencePiece GitHub. [Электронный ресурс]. – Режим доступа: <https://github.com/google/sentencepiece>
40. Sacremoses GitHub. [Электронный ресурс]. – Режим доступа: <https://github.com/alvations/sacremoses>
41. Hugging Face Accelerate. [Электронный ресурс]. – Режим доступа: <https://huggingface.co/docs/accelerate/index>
42. BitsAndBytes GitHub. [Электронный ресурс]. – Режим доступа: <https://github.com/TimDettmers/bitsandbytes>
43. microsoft/git-large-coco. Hugging Face Model. [Электронный ресурс]. – Режим доступа: <https://huggingface.co/microsoft/git-large-coco>
44. Helsinki-NLP/opus-mt-en-ru. Hugging Face Model. [Электронный ресурс]. – Режим доступа: <https://huggingface.co/Helsinki-NLP/opus-mt-en-ru>
45. TheBloke/Mistral-7B-Instruct-v0.2-GGUF. Hugging Face Model. [Электронный ресурс]. – Режим доступа: <https://huggingface.co/TheBloke/Mistral-7B-Instruct-v0.2-GGUF>
46. Национальная стратегия развития искусственного интеллекта на период до 2030 года.
47. Гражданский кодекс Российской Федерации. Часть четвертая от 18.12.2006 № 230-ФЗ (ред. от 01.07.2021).
48. Федеральный закон от 29.12.2010 № 436-ФЗ «О защите детей от информации, причиняющей вред их здоровью и развитию».

ПРИЛОЖЕНИЕ А

Apache License Version 2.0, January 2004 <http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensed by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works, a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

(e) You must not claim that Your Derivative Works constitute an official version of any components included in or used by the Work, including but not limited to the Python programming language, and You must clearly indicate that Your Derivative Works are modifications.

(f) You must preserve all copyright and other proprietary rights notices contained within any included software components, including but not limited to those specific to the Python programming language distribution.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademark Licenses. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file. **[ГИПОТЕТИЧЕСКОЕ ДОПОЛНЕНИЕ ДЛЯ УЧЕБНЫХ ЦЕЛЕЙ]** This prohibition extends to the use of trade names, trademarks, service marks, or product names associated with any components used by or included in the Work, including specifically the "Python"trademark, without explicit permission from their respective owners.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by

reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

ПРИЛОЖЕНИЕ Б



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

**Программное решение на основе искусственного
интеллекта для генерации смешных подписей к изображениям**

Студент: Епифановский Александр Денисович

Руководитель курсовой работы: Строганов Юрий Владимирович

Консультант: Москвичев Николай Владимирович

Консультант: Левиев Дмитрий Олегович

Консультант: Якуба Дмитрий Васильевич

Москва, 2025г

Цель и задачи работы

Цель: разработка программного решения, способного генерировать смешные подписи к изображениям.

Задачи:

- проанализировать предметную область
- спроектировать программное решение
- реализовать программное решение
- исследовать характеристики полученного решения
- определить правовой статус полученного решения

Введение в предметную область

Применение:

- создание контента для личных аккаунтов в социальных сетях
- вирусный маркетинг
- создание мемов для развлекательных платформ

Виды юмора

Amusement (Развлечение):

Пример: Наблюдать, как голубь пытается украсть хот-дог у зазевавшегося туриста.

Enjoyment (Наслаждение):

Пример: Слушать, как друг с юмором рассказывает о своих неловких приключениях в отпуске.

Fun (Веселье):

Пример: "- Вовочка, назови три вида динозавров. - Тираннозавр, птеродактиль и... и всёзавр!"

Pleasure (Удовольствие):

Пример: Смотреть выступление любимого стендап-комика, чьи шутки умны и вызывают искренний смех.

Absurdity (Абсурдность):

Пример: По стене ползет кирпич, а за ним — второй. "Подвинься!" — говорит первый. "Ты что, с ума сошел? — отвечает второй. — Мы же в разных измерениях!"

Irony (Ирония):

Пример: Попасть под ливень и сказать: "Отличная погода, как раз собирался принять душ".

Laughableness (Забавность):

Пример: Видео, где панда неуклюже скатывается с горки, кувыркаясь и размахивая лапами.

Ludicrousness (Нелепость):

Пример: Представить себе совещание директоров в серьезной компании, где все сидят в пижамах с динозаврами.

Ridiculousness (Смехотворность):

Пример: Человек в костюме банана, который поскользнулся на банановой кожуре.

Whimsicality (Причудливость):

Пример: "Если бы огурцы умели петь, они бы наверняка исполняли грустные баллады о своей скорой кончине в салате".

Виды юмора

Wittiness (Остроумие):

Пример: "Лучше молчать и казаться дураком, чем открыть рот и развеять все сомнения". (Марк Твен)

Wryness (Ироничность, противоречивость):

Пример: Получив очередную квитанцию за коммунальные услуги, с усмешкой сказать: "Как приятно знать, что мои деньги идут на такое важное дело, как золотые краны в мэрии".

Burlesque (Бурлеск):

Пример: Комическая опера, где боги Олимпа спорят о том, чья очередь мыть посуду.

Caricature (Карикатура):

Пример: Шарж на жадного человека, изображенного с огромными руками, которые загребают деньги.

Farce (Фарс):

Пример: Сцена, где герой прячет любовницу в шкаф, пока муж стучится в дверь, а в это время из-под кровати вылезает грабитель.

Jest (Шутка, острота):

Пример: "- Доктор, меня все игнорируют! - Следующий!"

Lampoon (Памфлет, злая сатира):

Пример: Едкая статья, высмеивающая чиновника за строительство "инновационной" детской площадки, состоящей из одной сломанной качели.

Parody (Пародия):

Пример: Фильм, в котором секретный агент "Джеймс Бонд 00,7" вместо Aston Martin ездит на стареньких "Жигулях" и постоянно попадает в бытовые неприятности.

Satire (Сатира):

Пример: "- Почему у нас такие хорошие дороги? - Чтобы чиновники могли быстрее добираться до своих вилл за границей" ..

Нормативная база

Национальная стратегия развития искусственного интеллекта на период до 2030 года (Утверждена Указом Президента РФ от 10.10.2019 № 490).

Гражданский кодекс Российской Федерации (Часть четвертая) от 18.12.2006 № 230-ФЗ.

Федеральный закон «Об информации, информационных технологиях и о защите информации» от 27.07.2006 № 149-ФЗ.

Федеральный закон «О противодействии экстремистской деятельности» от 25.07.2002 № 114-ФЗ.

Федеральный закон «О противодействии терроризму» от 06.03.2006 № 35-ФЗ.

Федеральный закон «О защите детей от информации, причиняющей вред их здоровью и развитию» от 29.12.2010 № 436-ФЗ.

Анализ аналогов

Название	Фокус	AI	Вход	Выход	Лицензия
Supermeme.ai	Генерация мемов	Да	Текст/Ключевые слова, Изображение	Мем	Лицензия сайта
Imgflip	Создание мемов и GIF	Да	Текст, Изображение	Мем, GIF	Лицензия сайта
Pincel	Генерация текста, подписей к изображениям	Да	Текст, Изображение	Описание, подпись	Лицензия сайта
Magic Hour AI	Генерация мемов	Да	Тема/Идея	Мем	Лицензия сайта
Funny caption generator	Генерация смешных подписей	Да	Изображение	Смешная подпись	Лицензия сайта
Pallyy	Генерация подписей	Да	Изображение, Тип подписи, Доп. пожелания	Подпись	Лицензия сайта
Captionslab (aigo.tools)	Генерация подписей	Да	Ключевые слова/Описание	Подпись	Лицензия сайта
Galaxy AI	Генерация подписей	Да	Изображение	Подпись	Лицензия сайта
WritingMate AI	Генерация подписей	Да	Изображение	Подпись	Лицензия сайта

Название	Фокус	AI	Вход	Выход	Лицензия
Gemini	Генерация различного контента	Да	Текст, Изображение	Текст, Изображение	-
Neural Joking Machine	Генерация смешных подписей	Да	Изображение	Смешная подпись	-
Oxford Humorous Image Captions	Генерация и понимание юмора	Да	Изображение	Смешная подпись	-
Humor in AI: Massive Scale Crowd-Sourced Preferences and Benchmarks for Cartoon Captioning	Генерация юмористических подписей к изображениям из мультфильмов	Да	Изображение	Смешная подпись	-
AI Humor Generation: Cognitive, Social and Creative Skills for Effective Humor	Генерация юмористических подписей	Да	Изображение	Смешная подпись	-
XMeCap: Meme Caption Generation with Sub-Image Adaptability	Генерация подписей к мемам	Да	Изображение	Подпись к картинке	-

Анализ аналогов

Название	Фокус	AI	Вход	Выход	Лицензия
ChatGPT	Генерация текста	Да	Фото, текст, описание	Текст	–
Show and Tell: A Neural Image Caption Generator	Генерация подписей к изображениям	Да	Изображение	Описание картинки	–
Image Caption Generator	Генерация подписей	Да	Изображение и тип подписи	Мем	Лицензия сайта
Junia AI Image Caption Generator	Генерация описания ищображения	Да	Изображение или ссылка на него в интернете, дополнительные пояснения	Описание изображения	Лицензия сайта
10015.io Image Caption Generator	Генерация описания и подписей для социальных сетей	Да	Изображение	Описание и подписи для социальных сетей	Лицензия сайта

ZapCap	Генерация смешной подписи по описанию	Да	Описание изображения	Подпись к медиафайлу	Лицензия сайта
Chat100 Photo Caption Generator	Генерация подписей к изображениям	Да	Изображени или описание изображения и стиль опдписи	Мем	Лицензия сайта
Simplified	Генерация подписей к изображениям в социальных сетях	Да	Изображение	Смешная подпись	Лицензия сайта
Captionslab	Генерация подписей к изображениям	Да	Текстовое описание	Мем	Лицензия сайта

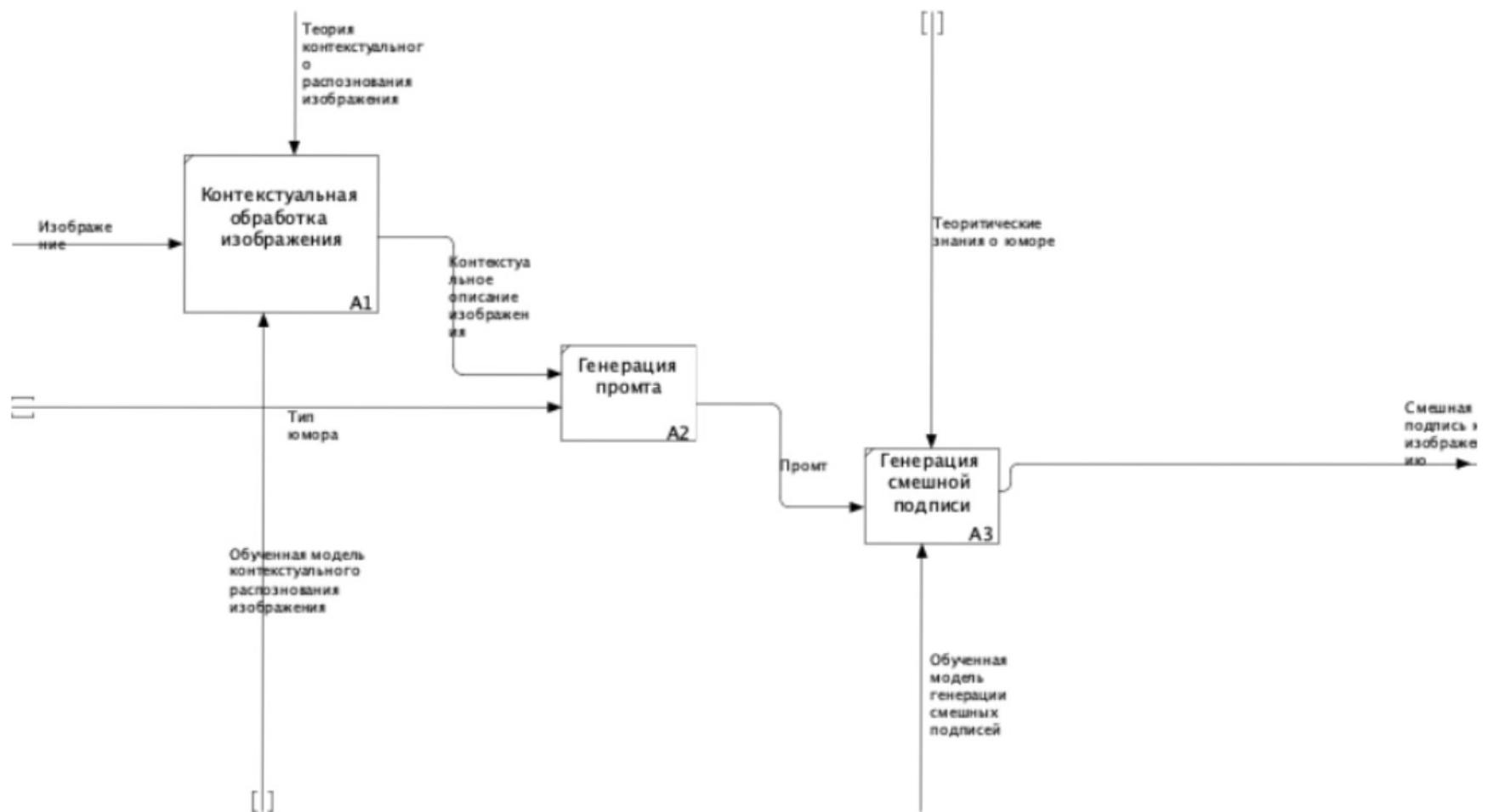
Анализ аналогов

Название	Фокус	AI	Вход	Выход	Лицензия
YesChat Image Caption Generator	Генерация подписей к изображениям	Да	Текстовое описание / Изображение	Картина с подписью	Лицензия сайта
Thread Creator Image Caption Generator	Генерация подписей	Да	Изображение	Подпись к изображению	Лицензия сайта
Videotok Caption Generator	Генерация подписей для социальных сетей	Да	Изображение и описание подписи (опционально)	Подпись для социальных сетей	Лицензия сайта

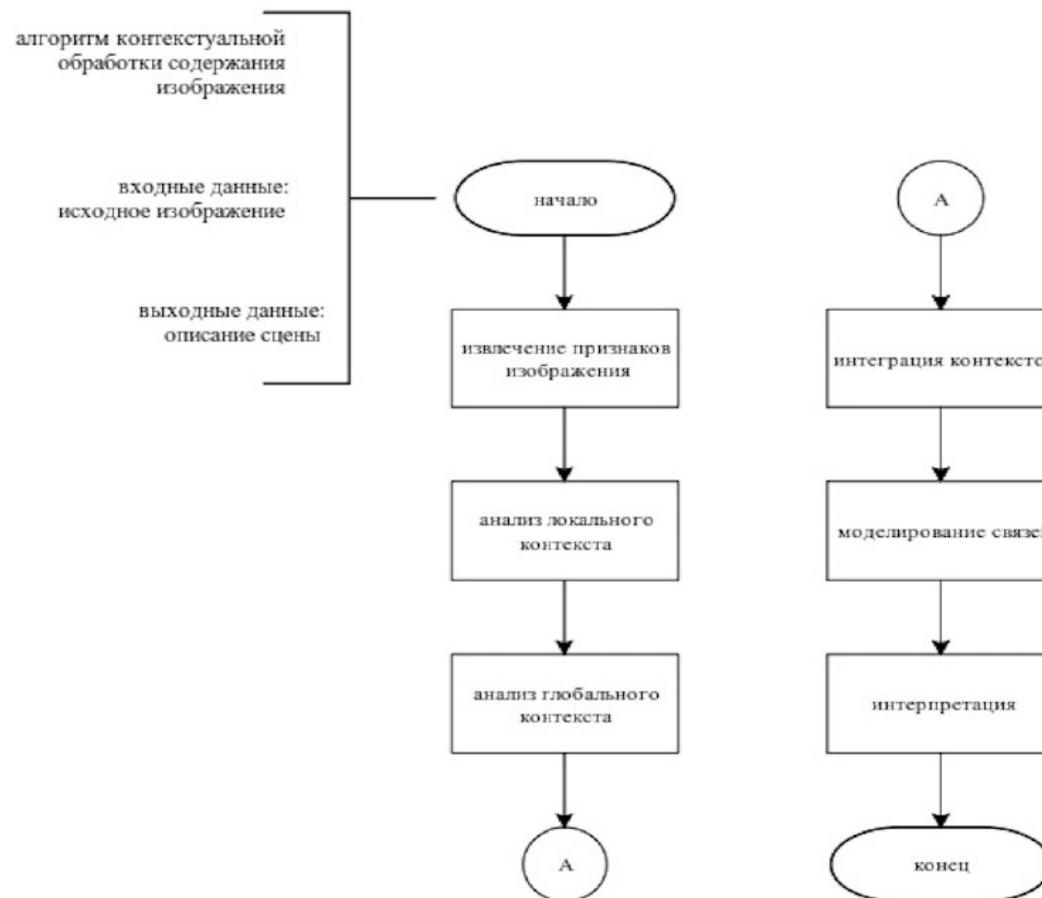
Ограничения и этические аспекты практической реализации

- На данный момент времени отсутствуют общедоступные и технически совершенные методы для надежного автоматического цензурирования контента пары «изображение – подпись»
- Практическая реализация такого продукта невозможна без предварительного решения фундаментальных проблем в области безопасного и этичного ИИ.

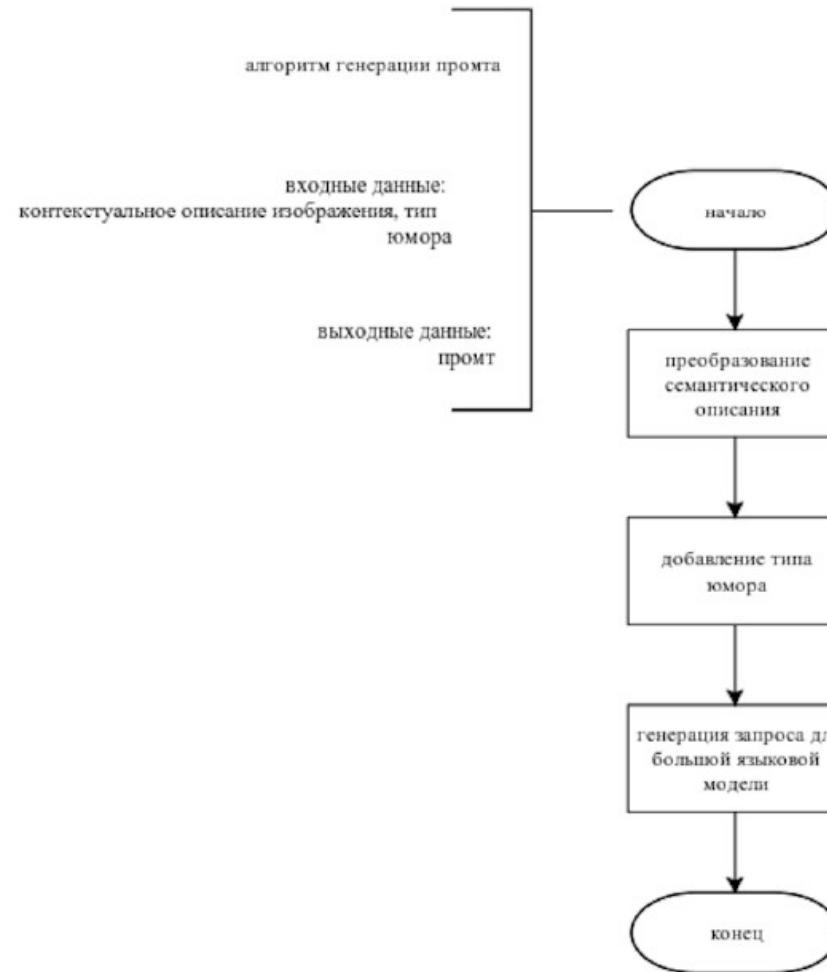
Функциональная диаграмма работы программы



Модуль контекстуальной обработки изображения (А1)



Модуль генерации промта (A2)



Модуль генерации смешной подписи (A3)

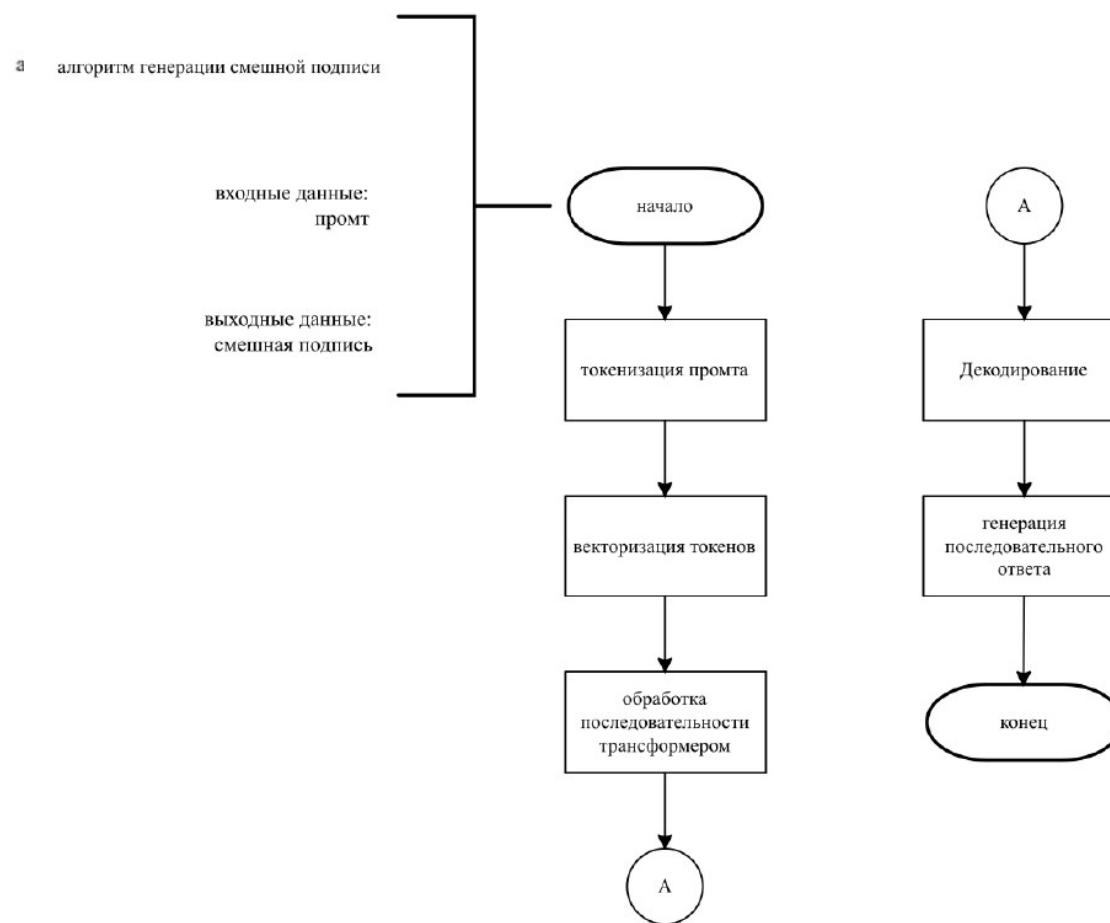
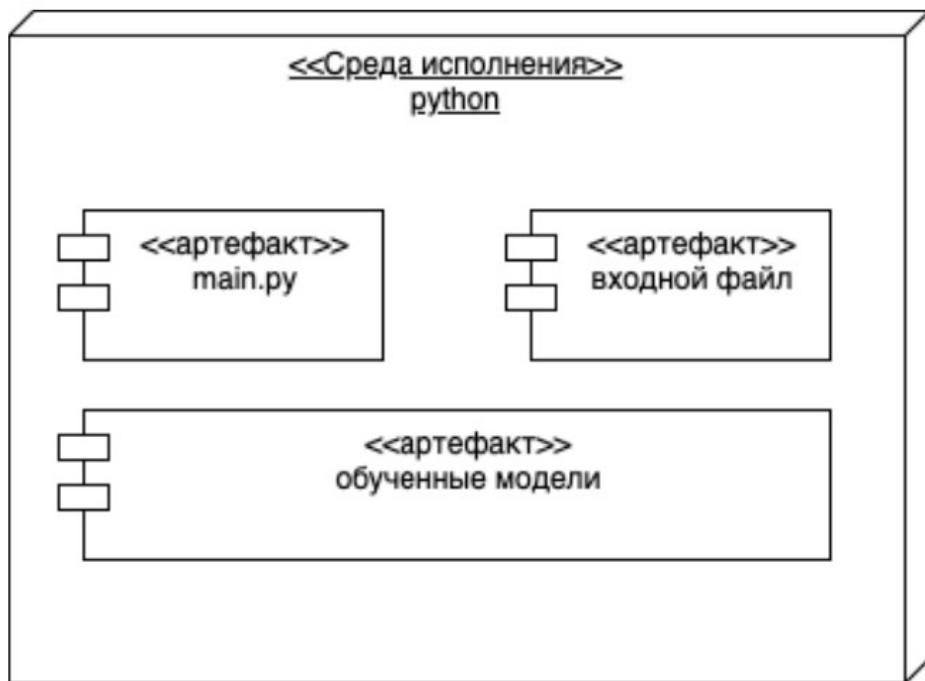


Диаграмма развертывания



Используемые
библиотеки:

- Pillow
- Transformers
- PyTorch
- SentencePiece
- Sacremoses
- Accelerate
- BitsAndBytes

Организационно-правовая часть

Цель ПО: Генерация юмористических подписей к изображениям с использованием нейронных сетей (анализ изображения + генерация текста).

Регулирование: Действующее законодательство Российской Федерации.

Нейронные сети и ИИ в законодательстве РФ:

1)Прямое юридическое определение "нейронной сети" отсутствует. Они считаются программно операционным комплексом и в соответствии с законодательством ответственность за использование лежит на предприятии и разработчике.

2)Технологии НС = ключевой компонент систем ИИ.

3)Развитие ИИ регулируется стратегическими документами (например, Национальная стратегия развития ИИ до 2030 г.).

4)ПО на основе НС = программа для ЭВМ (объект авторского права, ГК РФ Ч.4).

5)Использование ИИ для создания контента налагает обязательства по соблюдению законодательства (информация, данные, запрещенный контент).

Проблемы цензурирования и ограничения практического использования

Основная проблема: Отсутствие эффективных методов "жесткого" (гарантированного) цензурирования пар «изображение – текст».

Ограничения существующих фильтров (бан-листы):

- 1)Контекстуальная слепота (не понимают иносказаний, иронии).
- 2)Неспособность оценить семантическое соответствие и этичность (изображение + текст).
- 3)Вероятностная природа генерации НС (риск непредвиденного контента).

Необходимость экспертной оценки: Комплексная оценка вывода ПО требует человеческого анализа. Автоматизация на текущем уровне ненадежна.

Ключевой вывод по использованию:

- 4)Техническая реализация кода возможна.
- 5)НО: Реализация конечного продукта для широкого использования НЕВОЗМОЖНА.
- 6)Причина: Отсутствие гарантий предотвращения генерации текста, противоречащего законам РФ.

Возрастное ограничение: 18+ (согласно ФЗ № 436-ФЗ) из-за невозможности полной фильтрации контента.

Письмо с обращением для получения консультации по данному вопросу было отправлено в отдел судебной работы, начальнику судебной работы Житиной Ирине Викторовне 29 мая 2025 г.

Лицензия для ПО

Критерий	Apache 2.0	MIT	BSD-3-Clause	PSF
Коммерческое использование	Да	Да	Да	Да
Модификация	Да	Да	Да	Да
Распространение	Да	Да	Да	Да
Ограничение ответственности	Да	Да	Да	Да
Сохранение уведомления об авторских правах и тексте лицензии	Да	Да	Да	Да
Указание изменений	Да (значительные)	Нет (рекомендуется)	Нет (рекомендуется)	Да (в документации)
Предоставление исходного кода (при распространении бинарников)	Нет	Нет	Нет	Нет

Предоставление патентных прав	Да (явное)	Неявное	Неявное	Да (явное)
Использование товарных знаков	Нет (требуется разрешение)	Нет	Нет	Нет (требуется разрешение)
Сублицензирование	Да	Да	Да	Да
Копирайт (требование открытости производных работ)	Нет	Нет	Нет	Нет
Запрет на использование имени для продвижения (No Endorsement Clause)	Ограничено (товарные знаки)	Нет	Да (явный)	Ограничено (товарные знаки)
Совместимость с GPL	Да (с GPLv3+)	Да (с GPLv2+)	Да (с GPLv2+)	Да (с GPLv2+)
Сложность/Длина лицензии	Средняя	Очень короткая	Короткая	Средняя

Выбранная лицензия: Apache 2.0 с ограничениями PSF

Ограничения, накладываемые PSF

Запрет использования торговой марки Python без разрешения;

Запрет выдачи модифицированной версии, за официальный Python;

Сохранение уведомлений о правообладании в Python.

Заключение

Было разработано программное решение, способное генерировать смешные подписи к изображениям

Выполнены следующие задачи:

- проанализирована предметная область
- спроектировано программное решение
- реализовано программное решение
- исследованы характеристики полученного решения
- определен правовой статус полученного решения