

Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 23.М07-мм

Разработка web-приложения для разметки наборов данных

Коновалов Илья Олегович

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
Ассистент кафедры ИАС Г.А. Чернышев

Санкт-Петербург
2024

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
2.1. Обзор альтернативных решений	6
2.2. Технологии	9
3. Требования к функциональности	11
3.1. Страница со списком наборов данных	11
3.2. Страница набора данных	11
4. Реализация	13
4.1. Предметная область и модель данных	13
4.2. Серверная сторона	14
4.3. Клиентская сторона	16
5. Эксперимент	18
5.1. Набор данных для тестирования	18
5.2. Аппаратно-программная конфигурация	18
5.3. Методика эксперимента	19
5.4. Результаты эксперимента	19
Заключение	21
Список литературы	22

Введение

В современном мире значительная часть коммуникаций между людьми происходит посредством общения в социальных сетях, и в частности в мессенджерах [1]. Через мессенджеры проходит колоссальное количество информации, однако далеко не каждый человек готов тратить время и мыслительные ресурсы на перечитывать тысяч сообщений в чатах, накопившихся за день. В то же время, пропуск некоторой информации для конкретного человека может оказаться критичным и создать лишние проблемы.

В этой связи естественным образом возникает необходимость создания инструмента суммаризации чатов, способного делать ”выжимку” из накопившихся за некоторый период сообщений, чтобы человек мог получить самую необходимую информацию, не тратя на это много усилий.

Для реализации подобного инструмента могут быть задействованы такие методы, как: обработка естественного языка (natural language processing, NLP), машинное обучение (machine learning, ML), информационный поиск (information retrieval, IR). Для реализации этих методов необходимы размеченные данные, которые встречаются в открытом доступе довольно редко и зачастую являются специфичными для определенной задачи. К тому же работа с чатами требует особого подхода, поскольку тексты в них зачастую неполные, фрагментированные и содержат большое количество сленга или ошибок.

В работах [3] и [2] было представлено desktop-приложение под названием «Chat Corpora Annotator»¹, предназначенное для разметки чатов и реализованное на языке C#. В нем была реализована следующая функциональность:

- Загрузка наборов данных;
- Разметка наборов данных;
- Продвинутый поиск по наборам данных;

¹<https://github.com/yakovypg/Chat-Corpora-Annotator>

- Выгрузка размеченных наборов данных.

Однако, такое решение оказалось неприемлимым по следующим причинам:

- Отсутствие кроссплатформенности;
- Сложность интеграции инструментов машинного обучения, написанных на языке Python.

Таким образом возникла цель создания web-версии данного приложения, которое потенциально могло бы разрешить все обозначенные выше проблемы, при этом обеспечив оптимальный пользовательский опыт.

1. Постановка задачи

В предыдущей работе был реализован прототип на языке Python, который, обладал ограниченной функциональностью и низкой эффективностью при работе с большими наборами данных.

В этой связи целью данной работы является разработка нового более эффективного решения, которое расширяло бы функциональность предыдущей версии. Для её выполнения были поставлены следующие задачи:

1. Провести обзор альтернативных решений для разметки наборов данных;
2. Выбрать подходящие технологии, библиотеки и фреймворки для реализации решения;
3. Перенести ранее реализованную функциональность;
4. Повысить скорость индексации данных.

2. Обзор

2.1. Обзор альтернативных решений

На рынке существует множество инструментов для разметки текстовых наборов данных. В этом разделе будут приведены несколько наиболее популярных из них и проведен сравнительный анализ.

2.1.1. Prodigy

Prodigy² — инструмент для разметки различных видов данных (в том числе и текстовых) от создателей библиотеки spaCy. Поддерживает обширный набор инструментов для разметки и продвинутого анализа, такие как: распознавание именованных сущностей (Named Entity Recognition, NER) и отношений между ними, классификацию документов и в том числе автоматизированную разметку на основе моделей spaCy. Prodigy предназначен для небольших команд, в связи с чем не предоставляет SaaS услуг и развертывается локально. Исходные коды закрыты, продукт распространяется только платно.

2.1.2. LightTag

LightTag³ — минималистичный инструмент для коллаборативной разметки данных для задач NLP. Поддерживает распознавание именованных сущностей, отношения между сущностями, категоризацию интервалов, классификацию документов и автоматизированную разметку ML-моделями. Поддерживает как SaaS, так и on-premise модель распространения. Исходные коды закрыты и для организаций сервис является платным, однако для индивидуального использования в образовательных целях сервис бесплатен.

²<https://prodi.gy>

³<https://www.lighttag.io>

2.1.3. LabelBox

LabelBox⁴ — многопользовательский инструмент для разметки наборов данных, предназначенный для машинного обучения и позволяющий работать с различными форматами данных, включая текстовые. Поддерживает распознавание именованных сущностей, отношения между сущностями, классификацию документов и автоматизированную разметку ML-моделями. Продукт является проприетарным и предоставляется по модели SaaS бесплатно для индивидуального использования и платно для организаций.

2.1.4. TagTog

TagTog⁵ — один из первых коммерческих решений для разметки текстовых наборов данных, создававшийся для обработки медицинских данных [4]. Он поддерживает многопользовательскую разметку, распознавание именованных сущностей, отношения между ними, а также интеграцию с ML-моделями для автоматизированной разметки. Возможны использование как в облаке, так и on-premise. Распространяется бесплатно для индивидуального использования и платно для организаций.

2.1.5. Generative AI Lab

Generative AI Lab⁶ — no code NLP платформа, позволяющая осуществлять многопользовательскую разметку текстовых данных. Поддерживает распознавание именованных сущностей, отношения между сущностями и автоматическую разметку с использованием моделей из библиотеки Spark NLP. Разворачивается на платформах популярных облачных провайдеров или локально (on-premise). Сам по себе продукт бесплатен, но необходимо оплачивать инфраструктуру, предоставляемую облачным провайдером.

⁴<https://labelbox.com>

⁵<https://docs.tagtog.com>

⁶<https://nlp.johnsnowlabs.com/docs/en/alab/quickstart>

2.1.6. LabelStudio

LabelStudio⁷ — инструмент для разметки различных типов данных: в области текстовой разметки поддерживает гибкую настройку схемы под конкретную задачу, распознавание именованных сущностей, отношений, классификацию документов и подключение ML-моделей для автоматической разметки. Имеется возможность как для локального развертывания, так и использования облачной платформы. Распространяется в виде двух версий: базовой, которая является открытой и бесплатной, и промышленной, в которой имеется дополнительная функциональность по отношению к базовой версии.

2.1.7. Doccano

Doccano⁸ — инструмент для разметки текстовых данных с открытым исходным кодом. Поддерживает распознавание именованных сущностей, и автоматизированную разметку с использованием внешних API. Поддерживает развертывание как локально, так и в облачной инфраструктуре.

2.1.8. Выводы

Таким образом, большинство популярных инструментов либо имеют обширное количество функциональности для разметки и аналитики данных, но являются проприетарными, либо они открыты, но не поддерживают такой же широкий спектр возможностей. Кроме того, ни один из инструментов не поддерживает необходимого визуального интерфейса построочного аннотирования, что делает их малоприспособленными для решения задачи разметки наборов данных на основе чатов.

Мы пытаемся создать инструмент, который был бы открытым, но в то же время эффективным и разнообразным в плане предоставляемых возможностей по аналитике данных.

⁷<https://labelstud.io>

⁸<https://doccano.github.io/doccano>

2.2. Технологии

Предыдущая версия приложения основывалась на стратегии server-side-rendering (SSR), что значительно ограничивало гибкость и масштабируемость системы. Исходя из этого было принято решение разделить приложение на SPA (single page application) клиент и сервер, реализующий REST API (representation state transfer application programming interface).

2.2.1. Серверная сторона

Для написания серверной стороны приложения использовался язык Java, в связке со Spring Framework⁹. Язык Java был выбран в силу того, что он наилучшим образом подходит для написания больших приложений, в которых требуется обрабатывать большие объемы данных.

Spring Framework был выбран по следующим причинам:

- Встроенный механизм инверсии управления (IoC), который помогает упростить разработку приложений путем снижения связанности компонентов и повышения их переиспользуемости;
- Модульная архитектура, позволяющая использовать только необходимые модули и интегрировать их в программные продукты по мере необходимости;
- Поддержка аспектно-ориентированного программирования, упрощающего реализацию функциональности, не относящейся к бизнес-логике приложения, такой как транзакционное управление или логирование.

В приложении использовались следующие модули данного фреймворка:

- Spring Boot — модуль, упрощающий процесс конфигурации проекта и управления зависимостями;

⁹<https://docs.spring.io/spring-framework/reference/index.html>

- Spring Web — модуль, предоставляющий инструменты для создания веб-приложений;
- Spring Data JPA — модуль, предоставляющий ORM решение согласно JPA спецификации¹⁰ (в данном случае Hibernate¹¹).

В роли хранилищ выступают следующие базы данных:

- PostgreSQL для хранения метаданных;
- Elasticsearch для наборов данных, загружаемых пользователями.

ElasticSearch — это распределенная поисковая и аналитическая система, специализирующийся на поиске и агрегации больших объемов данных в реальном времени. На практике представляет из себя сервер, предоставляющий REST API для загрузки и извлечения данных¹².

PostgreSQL — это реляционная СУБД¹³.

2.2.2. Клиентская сторона

Для реализации клиентской стороны использовался язык JavaScript в связке с фреймворком Vue.js¹⁴, на основе которого было создано SPA.

Vue.js был выбран по следующим причинам:

- Легковесность;
- Простота использования;
- Высокая производительность.

В качестве библиотеки компонентов использовался Vuetify¹⁵, который предоставляет обширный набор Vue компонентов в стилистике material design.

¹⁰<https://jakarta.ee/specifications/persistence>

¹¹<https://hibernate.org/orm>

¹²<https://www.elastic.co/elasticsearch>

¹³<https://www.postgresql.org>

¹⁴<https://vuejs.org>

¹⁵<https://vuetifyjs.com>

3. Требования к функциональности

С учетом предыдущих наработок, для новой версии были сформулированы нижеперечисленные требования, которым должно удовлетворять приложение.

3.1. Страница со списком наборов данных

Данная страница предусматривает следующие действия:

- Загрузка набора данных;
- Переименование и удаление наборов данных;
- Сортировка и фильтрация по названию;
- Выгрузка размеченных наборов данных в форматах CSV и JSON.

3.2. Страница набора данных

Данная страница состоит из двух вкладок, каждая из которых отображает таблицу с данными. Для каждой из таблиц должна быть предусмотрена возможность включения/исключения, перестановки и изменения ширины столбцов. Также необходимо реализовать поддержку произвольной схемы наборов данных, отображаемых в таблицах.

3.2.1. Вкладка разметки

- Постраничный просмотр набора данных;
- Разметка строк;
- Создание, переименование, удаление, сортировка, фильтрация меток.

3.2.2. Вкладка поиска

- Поиск по запросу;
- Постраничный просмотр результатов поиска.

4. Реализация

4.1. Предметная область и модель данных

Предметная область приложения из следующих сущностей:

- Dataset — метаинформация о наборах данных;
- Label — информация о метках, присваиваемых строкам набора данных в ходе разметки;
- Annotation — сопоставление метки конкретной строке.

Между наборами данных и метками установлено отношение один-ко-многим, то есть один набор данных может иметь несколько меток, в то время как каждая метка принадлежит единственному набору. В рамках одного набора данных все метки должны быть уникальны.

Также в базе данных сохраняется информация о хранилищах, концепция которых будет описана ниже.

Таким образом, схема базы данных отображена на Рисунке 1.

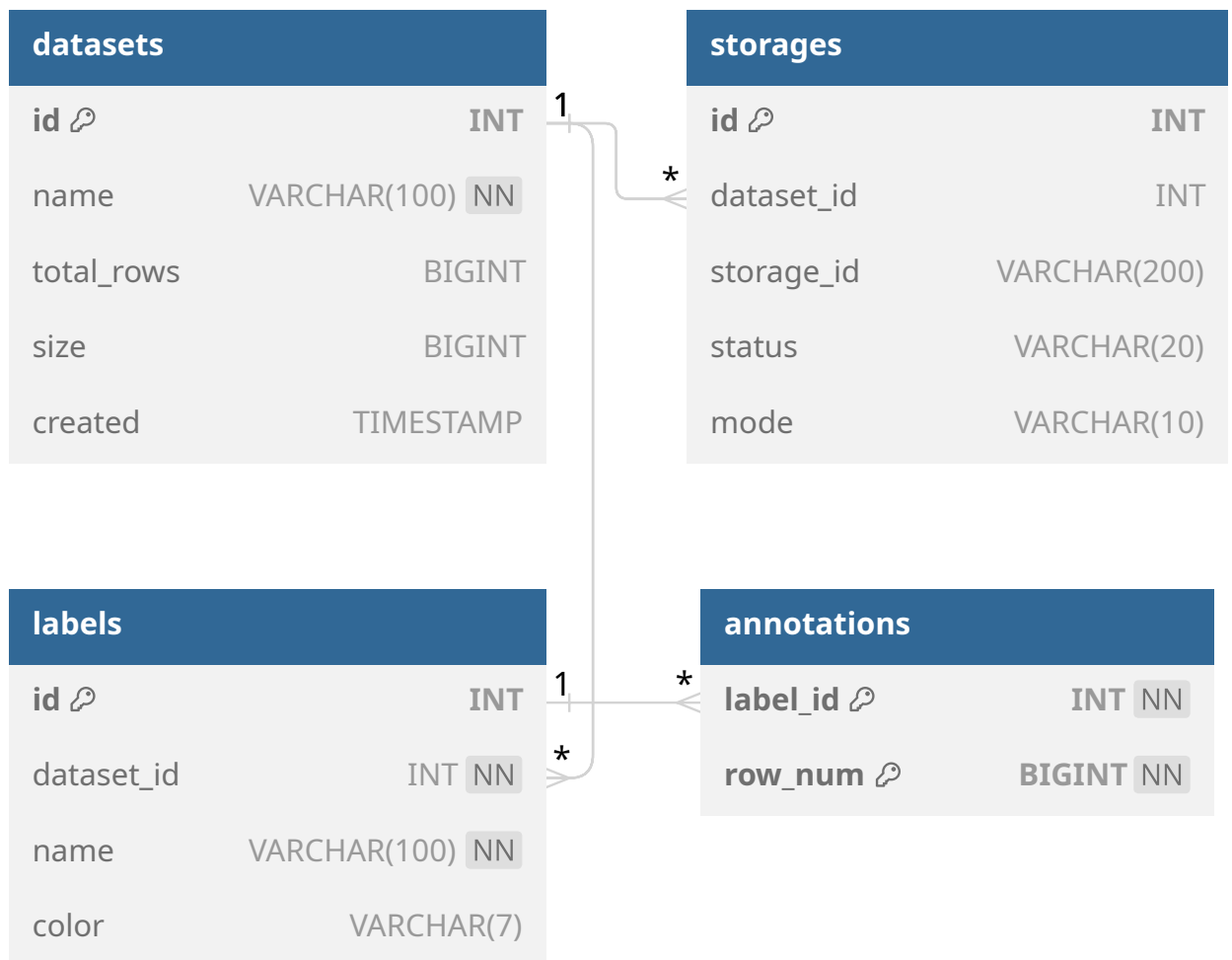


Рис. 1: Схема базы данных

4.2. Серверная сторона

4.2.1. Хранилища данных

Одним из главных нефункциональных требований к приложению была возможность обработки больших наборов данных (до 5 ГБ) без значительных затрат оперативной памяти. Другими словами, необходимо было добиться возможности обработки наборов данных без полной выгрузки в оперативную память. Для решения этой проблемы использовался потоковый подход с использованием Java Stream API, позволяющий обрабатывать большие объемы данных в последовательном режиме.

Поскольку индексация больших наборов данных может занимать значительное время, для того, чтобы обеспечить положительный поль-

зовательски опыт, необходимо было обеспечить возможность как можно быстрее начинать разметку не дожидаясь момента завершения индексации. Для решения этой проблемы были введены понятия первичного и вторичного хранилищ.

- Первичное хранилище — хранилище данных, поддерживающее операции извлечения данных и поиска. В данном случае первичным хранилищем является сервер Elasticsearch;
- Вторичное хранилище — хранилище данных, в которое данные загружаются относительно быстро, и которое поддерживает операции извлечения данных, но не поиска.

Система устроена таким образом, что сначала данные загружаются во вторичное хранилище, чтобы пользователь имел возможность начать размечать данные как можно быстрее. После завершения загрузки во вторичное хранилище, начинается миграция данных в первичное хранилище. После завершения миграции вторичное хранилище удаляется, и запросы на извлечение и поиск данных перенаправляются первичному хранилищу.

В процессе миграции в базу данных проставляются соответствующие статусы, что обеспечивает возможность мониторинга прогресса.

4.2.2. Поиск

Поиск по набору данных становится доступен с момента завершения миграции данных в первичное хранилище. По-умолчанию индексируются все поля набора данных и по ним выполняется полнотекстовый поиск. Также при поиске доступен простой язык запросов¹⁶, поддерживаемый Elasticsearch.

В качестве результата возвращается найденное сообщение (специальные символы экранированы) с выделенными совпадениями в тексте сообщения. Выделение происходит с помощью HTML тега ``.

¹⁶<https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-simple-query-string-query.html>

4.3. Клиентская сторона










Приложение состоит из нескольких страниц, которые были перечислены в пункте 3. Графический интерфейс данных страниц представлен на Рисунках 2, 3 и 4.

CCA

Datasets

Q Search

UPLOAD

Name ↑	Size	Rows	Created	
dataset_1	9.33 MB	100000	01/06/2024, 13:34:46	  
dataset_2	1.04 MB	11265	31/05/2024, 17:32:32	  
dataset_3	1.04 MB	11265	01/06/2024, 13:35:15	  

Items per page: 10 1-3 of 3

Рис. 2: Страница со списком наборов данных

CCA

ANNOTATE SEARCH

	#	Labels	Username	Text
<input type="checkbox"/>	1		sircharleswatson	no legumes either DATASET V2
<input type="checkbox"/>	2		janetwalters008	That bullet proof coffee sounds insane.
<input type="checkbox"/>	3		janetwalters008	That guy has huge eyes.
<input type="checkbox"/>	4	smile user-tag	sircharleswatson	@janetwalters008 It is. but it works. some people just can't handle the taste :P
<input type="checkbox"/>	5		phgilliam	They guy that came up with the idea is kind of a joke though...
<input type="checkbox"/>	6	smile	odrisk	that sounds like torture actually :)
<input type="checkbox"/>	7		phgilliam	the*
<input type="checkbox"/>	8		janetwalters008	I might try it out for fun just one bullet proof coffee that is.
<input checked="" type="checkbox"/>	9	user-tag	sircharleswatson	@phgilliam I agree. he's pretty extreme lol
<input type="checkbox"/>	10		sircharleswatson	he's like the Bear Grylls of diets
<input type="checkbox"/>	11		sircharleswatson	haha
<input type="checkbox"/>	12		odrisk	I have zero intention of doing the whole diet bit of it, I just want the nommy creamy fatty coffee
<input type="checkbox"/>	13		odrisk	and the energy
<input checked="" type="checkbox"/>	14		sircharleswatson	I can't help but laugh at my own joke/reference lol
<input type="checkbox"/>	15		sircharleswatson	Anyone near LA/Santa Monica, CA want to host me and my wife for a week or two? :D

Labels

Current label: lol

Q Search ADD

Name

☒ lol

☐ user-tag

☐ smile

ANNOTATE

Columns

Show row numbers

Name	Width
labels	0
username	0
text	600

Excluded

Name	Width
sent	0

Рис. 3: Разметка набора данных

CCA

ANNOTATE

SEARCH

Q hello

SEARCH

Username	Text
RussEby	Hello @mykey007
AhsanBudhani	hello @mkey007
alexung	Hello world
odrick	hello @officialswanson
	hello world!
officialswanson	Hello campers. :P
AhsanBudhani	Hello Everyone I am new here
theepdinker	hello world! Newby to Free Code Camp here.
adnhit	Hello all! I was away on vacation for a while there
JohnQQ	@Kadams223 Hello, first of all Happy new year!, secondly, you need to add another (background-size: 75px 150px); to each @keyframe.

Items per page: 50 1-10 of 10 |< < > >|

Columns

Show row numbers

Included

Name	Width
username	0
text	600

Excluded

Name	Width
sent	0

Рис. 4: Поиск по набору данных

5. Эксперимент

В прошлой работе проводилось тестирование приложения на предмет времени индексации загруженного набора данных с использованием библиотеки Whoosh ¹⁷. Данный эксперимент проводился с целью выяснения возможностей масштабируемости приложения для работы с крупными наборами данных.

Ниже приведен повтор данного эксперимента для новой версии системы, где используется Elasticsearch.

5.1. Набор данных для тестирования

Для тестирования использовался набор данных, содержащий сообщения онлайн-курса по обучению программированию¹⁸.

Предварительно была произведена следующая обработка:

1. Удаление лишних столбцов;
2. Удаление записей, содержащих нулевые значения хотя бы в одном из столбцов;
3. Переименование столбцов для приведения к требуемому формату,

По итогу были получен набор данных размером 500 МБ, содержащий 5037827 записей.

Средняя длина сообщения составила 60 символов ($\sigma = 100$). При таких условиях часть набора данных размером 1 МБ будет включать приблизительно 10000 записей.

5.2. Аппаратно-программная конфигурация

Тестирование производилось на устройстве Macbook Air 13, обладающем следующими характеристиками:

- Процессор: Apple Silicon M1

¹⁷<https://whoosh.readthedocs.io/en/latest>

¹⁸<https://www.kaggle.com/datasets/freecodecamp/all-posts-public-main-chatroom>

- ОЗУ: 8 ГБ
- ОС: MacOS Ventura 13.4
- Размер SSD диска: 256 ГБ

Эксперимент запускался в виртуальной машине со следующими характеристиками:

- Кол-во ядер процессора: 6
- ОЗУ: 4 ГБ
- ОС: Ubuntu 22.04
- Дисковое пространство: 50 ГБ
- Версия Python: 3.11

5.3. Методика эксперимента

Для эксперимента использовалось 5 наборов данных размерами 1 МБ, 10 МБ, 100 МБ, 500 МБ, 1000 МБ, полученных из исходного за счет усечения или повторения. В ходе эксперимента измерялись время индексирования и размер полученного индекса. Для каждого набора данных выполнялось 5 запусков, итоговые результаты усреднялись.

Время замерялось с учетом загрузки данных на сервер Elasticsearch, поскольку индексация производится одновременно с загрузкой. Загрузка производилась в двух потоках по 10000 тысяч записей за запрос, что соответствует параметрам загрузки, установленным в приложении по-умолчанию.

5.4. Результаты эксперимента

По итогам эксперимента были получены результаты, представленные в Таблице 1.

Таблица 1: Производительность Elasticsearch

Размер набора данных, МБ	Время индексации, с	Размер индекса, МБ
1	2.1 ± 0.3	2.3 ± 0.2
10	7.3 ± 0.5	21.3 ± 2.1
100	35.1 ± 5.3	230.2 ± 30.6
500	203.0 ± 17.2	1250.6 ± 150.2
1000	417.8 ± 21.4	2013.0 ± 284.6
5000	2329.0 ± 97.4	9932.6 ± 984.5

Результаты эксперимента из предыдущей работы представлены в Таблице 2.

Таблица 2: Производительность Whoosh

Размер набора данных, МБ	Время индексации, с	Размер индекса, МБ
1	3.2 ± 0.2	4.6 ± 0.3
10	38.5 ± 2.0	33.2 ± 3.2
100	402.6 ± 23.3	298.7 ± 28.4
500	3978.0 ± 199.3	$12\,632.0 \pm 127.0$

По полученным результатам можно судить о том, что производительность индексации значительно увеличилась, а также о том, что Elasticsearch хорошо подходит для обработки крупных наборов данных, поскольку время индексации и память, занимаемая индексом, возрастают практически линейно относительно роста размера набора данных.

Заключение

По итогам работы выполнены следующие задачи:

- Проведен обзор альтернативных решений;
- Выбраны технологии для реализации;
- Перенесена старая и реализована новая функциональность;
- Значительно увеличена скорость индексации данных;

Направления для дальнейшей работы:

- Добавление поддержки многопользовательской разметки;
- Добавление поддержки продвинутых инструментов для поиска и анализа данных;
- Улучшение UI и UX.

Исходный код можно найти в GitHub репозиториях [TurboGoose/cc-backend](#) и [TurboGoose/cc-frontend](#).

Список литературы

- [1] Digital 2024 Global Overview Report. — URL: <https://wearesocial.com/us/blog/2024/01/digital-2024-5-billion-social-media-users/> (дата обращения: 1 июня 2024 г.).
- [2] Query Processing and Optimization for a Custom Retrieval Language / Yakov Kuzin, Anna Smirnova, Evgeniy Slobodkin, George Chernishev // Proceedings of the First Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning / Ed. by Laura Chiticariu, Yoav Goldberg, Gus Hahn-Powell et al. — Gyeongju, Republic of Korea : International Conference on Computational Linguistics, 2022. — Oct.. — P. 61–70. — URL: <https://aclanthology.org/2022.pandl-1.8>.
- [3] Smirnova Anna, Slobodkin Evgeniy, Chernishev George. [Situation-Based Multiparticipant Chat Summarization: a Concept, an Exploration-Annotation Tool and an Example Collection](#) // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop / Ed. by Jad Kabbara, Haitao Lin, Amandalynne Paullada, Jannis Vamvas. — Online : Association for Computational Linguistics, 2021. — Aug.. — P. 127–137. — URL: <https://aclanthology.org/2021.acl-srw.14>.
- [4] tagtog: interactive and text-mining-assisted annotation of gene mentions in PLOS full-text articles / Juan Miguel Cejuela, Peter McQuilton, Laura Ponting et al. // [Database](#). — 2014. — 04. — Vol. 2014. — P. bau033. — <https://academic.oup.com/database/article-pdf/doi/10.1093/database/bau033/8245396/bau033.pdf>.