Сервис поиска соседних объектов

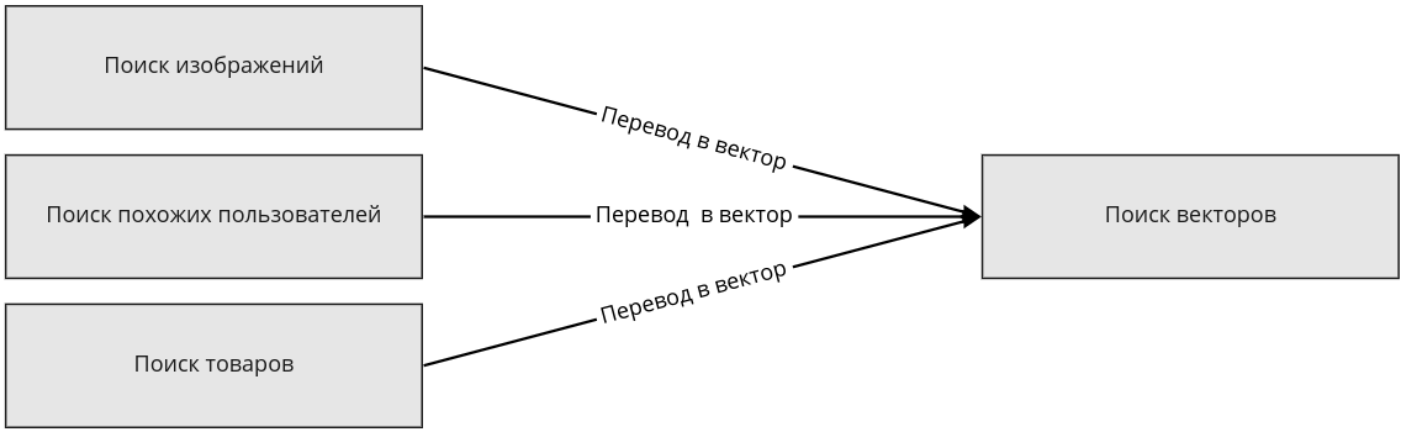
Выполнил: Амеличев Константин, ПМИ 195

Научный руководитель: Тощев Андрей Александрович, Tinkoff.ru

Цель работы

Создание «универсального» поискового сервиса SimSearch (от англ. *similarity search*).

Поисковый объект — что угодно, что можно векторизовать с сохранением свойств.



Требования к поисковому сервису

- Гибкость при решении поисковой задачи
- Поддержка различных баз: как статических, так и динамически меняющихся
- Быстрые ответы на запросы, даже при больших базах
- Возможность масштабировать сервис

SimSearch: актуальность и значимость

Подобного сервиса в открытом доступе нет, только продукты, заточенные под конкретную поисковую задачу

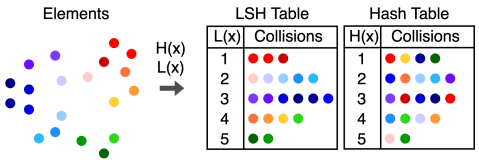
Преимущества SimSearch:

- Гибкость, которой нет у специализированных сервисов
- Универсальность: в предлагаемую модель обобщается множество поисковых задач
- Продвинутые применения: Multimodal space (поиск картинок по текстовому описанию)

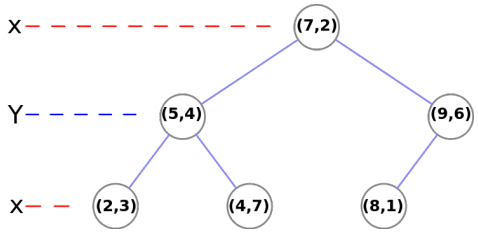
Поставленные задачи

- Изучение задачи поиска ближайших векторов (ANN: *approximate nearest neighbors*)
- Проектирование SimSearch
- Разработка и тестирование SimSearch
- Разработка поиска похожих изображений на основе SimSearch

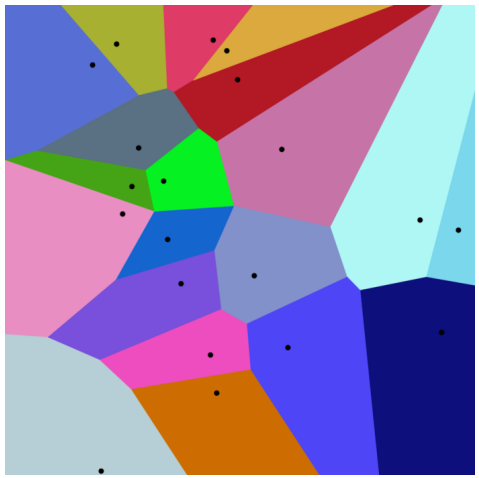
Задача ANN: изученные подходы



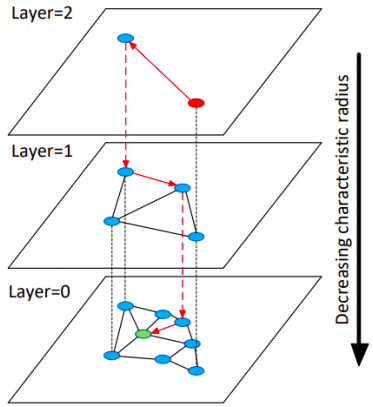
LSH



KD-tree



Quantization



HNSW

Вывод из исследования: приоритетные подходы — LSH и HNSW.

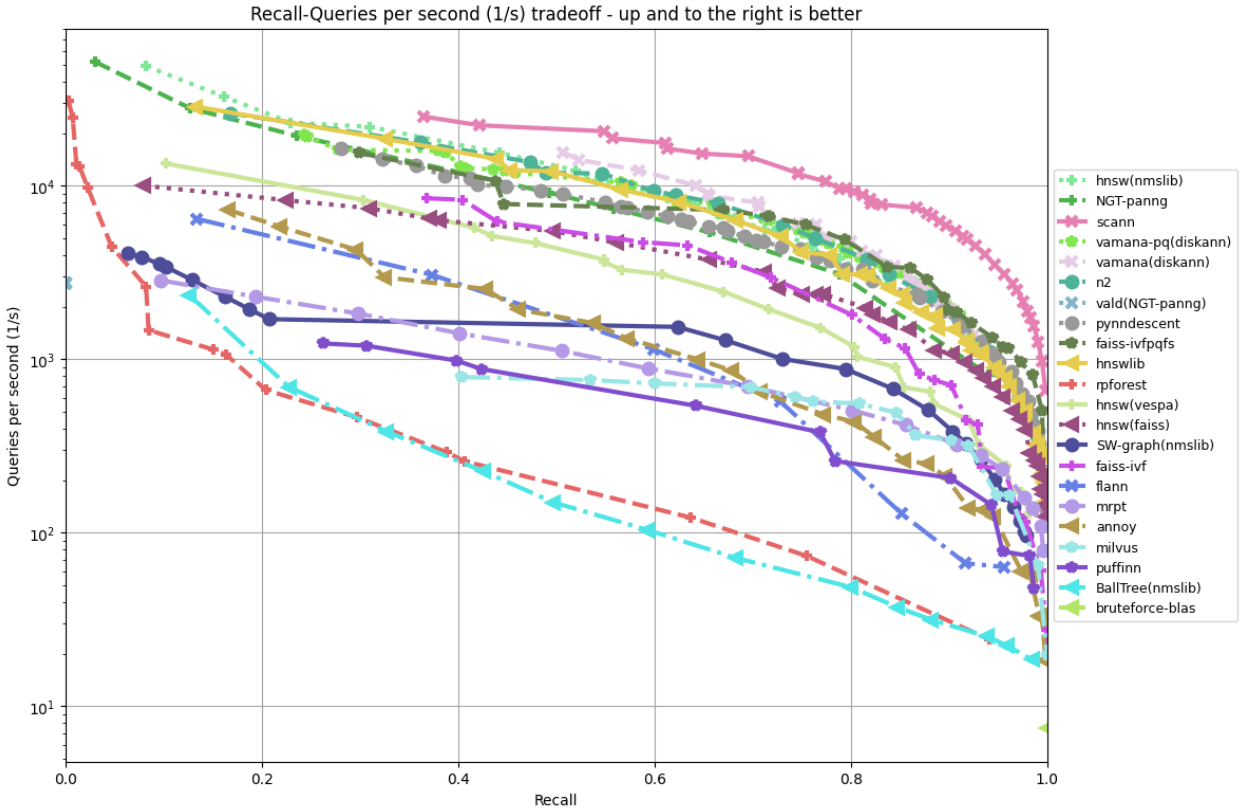
Задача ANN: Выбор библиотеки для LSH

Выбор из open-source библиотек на C++:

- LSHKit
- MyLSHBox
- slash

Выбор: LSHKit для алгоритма LSH — больше всего поддерживаемых расстояний и функций LSH, есть multi-probe lsh, lsh forest.

Задача ANN: Выбор библиотеки для HNSW

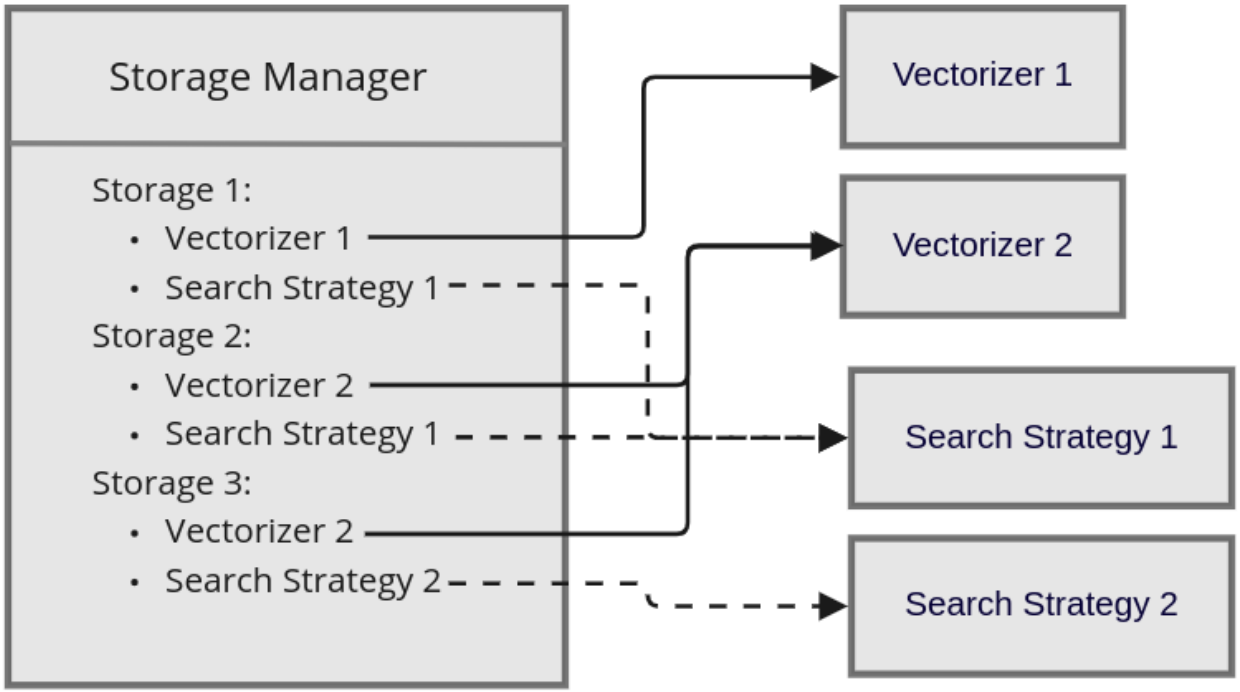


Тестирование на данных glove-100-angular (k = 10) [1]

Выбор: FaiSS для алгоритма HNSW — хорошая скорость, содержит также другие алгоритмы (например, на основе квантизации).

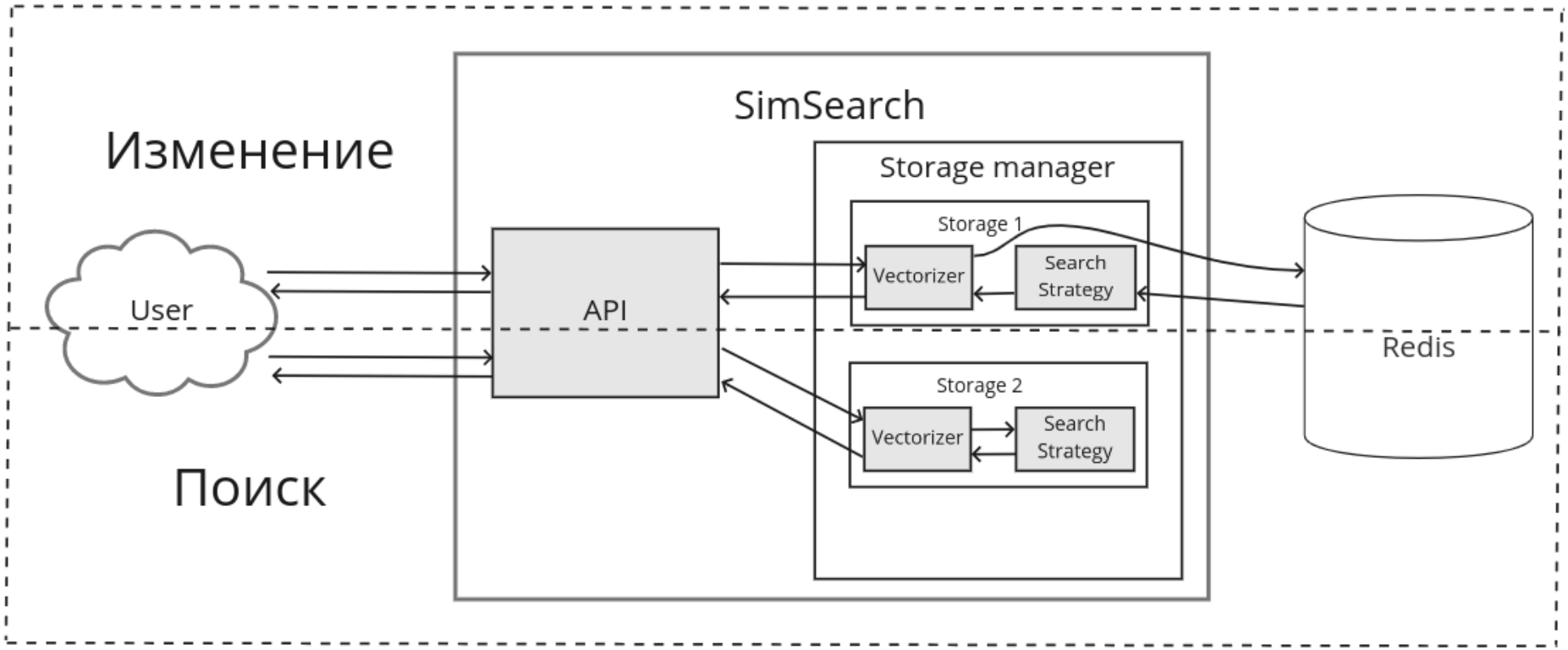
[1] ann-benchmark.com

SimSearch: проектирование

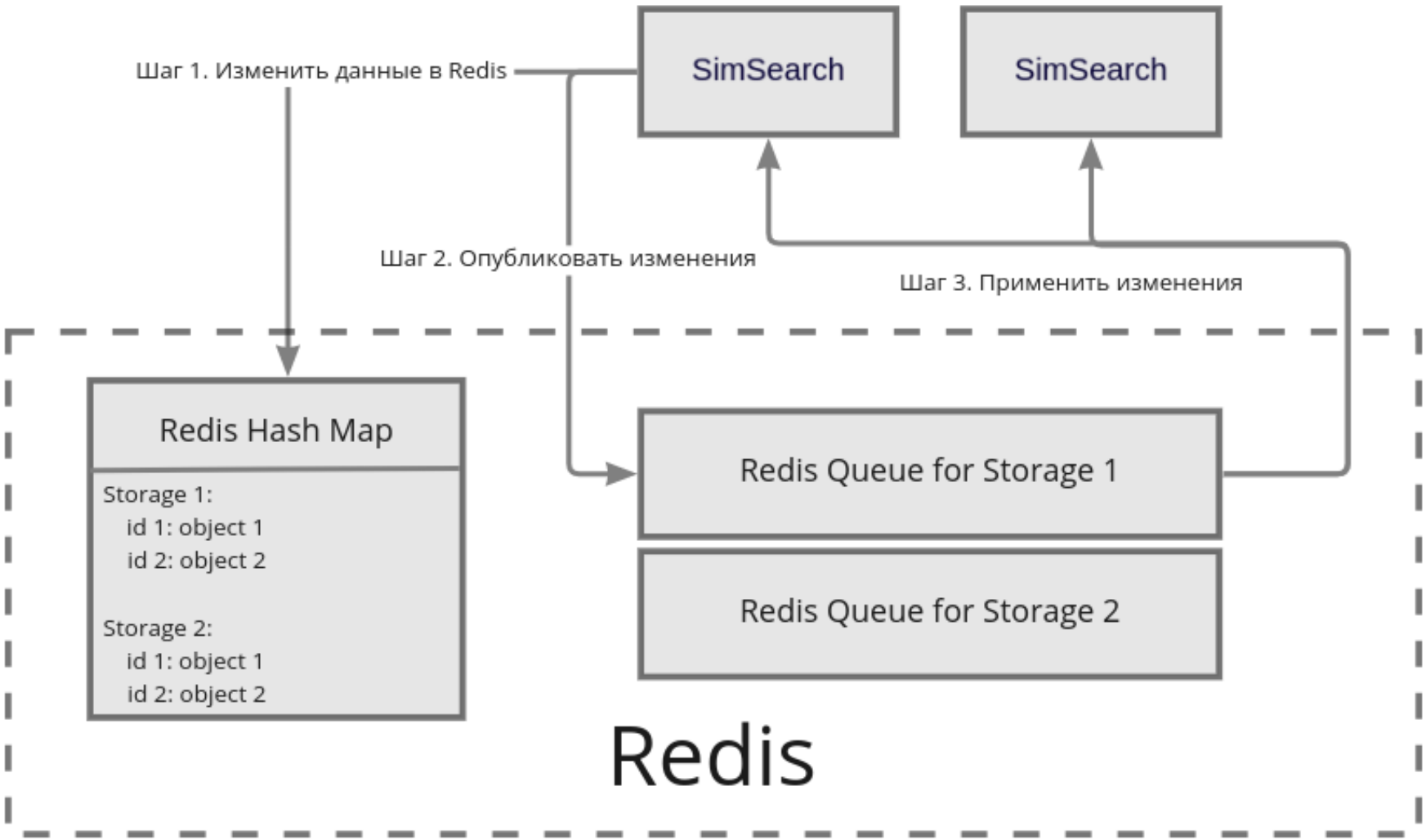


Выделяем векторизацию и стратегию поиска в разные модули. Благодаря общему интерфейсу модули можно комбинировать и эффективно переиспользовать!

SimSearch: архитектура



SimSearch: масштабируемость, Redis



SimSearch: тестирование

Стратегия	Размер хранилища	t(Add), мс	t(Search), мс
linear	1000	12	98
linear	5000	11	410
lshkit	1000	89	14
lshkit	5000	280	25
faiss-hnsw	1000	12	43.5
faiss-hnsw	5000	14	44

Вывод: для статической поисковой задачи лучше использовать стратегию с lshkit, для динамической — faiss-hnsw.

SimSearch: что еще?

- Cmake: сборка
- Docker: контейнеризация
- Pytest: тесты
- CI/CD в Gitlab

Поиск похожих изображений

Приложение для демонстрации возможностей SimSearch. Разработано на flask, взаимодействует с API SimSearch.

SimSearch

Отправить

Query:

Results:

Поиск похожих изображений: еще примеры

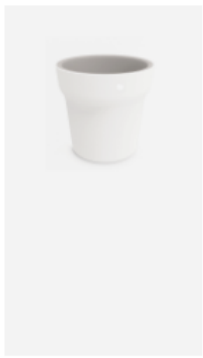
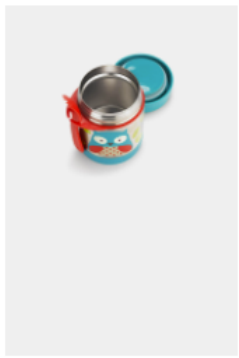
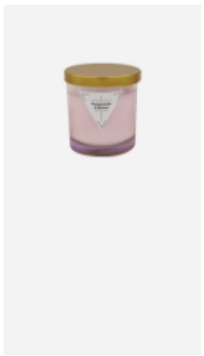
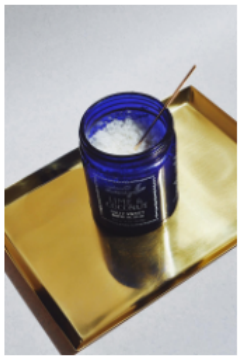
SimSearch

Отправить

Query:

Results:



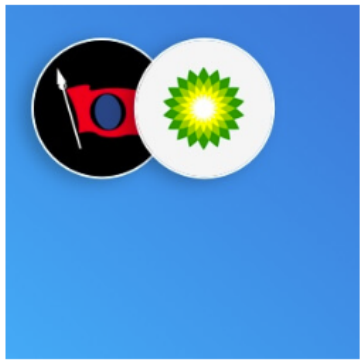

Поиск похожих изображений: неудачные примеры

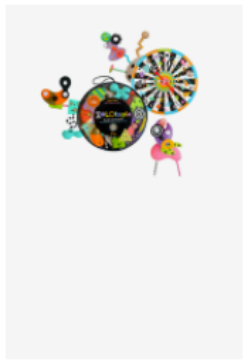
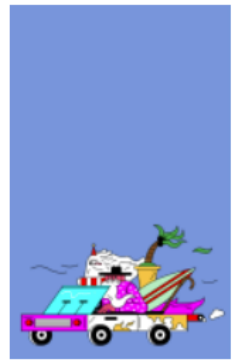
SimSearch

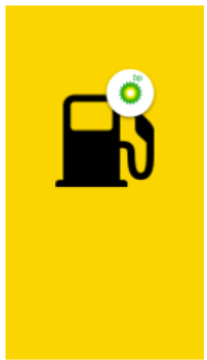
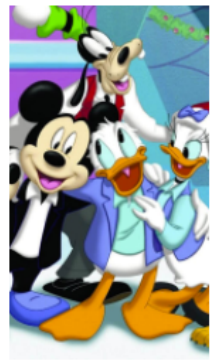
Отправить

Query:

Results:



Узнали, что логотип топливной компании напоминает ромашку :).

Результаты

- Сформулированы требования к «универсальному» поисковому сервису
- Проанализированы подходы к решению задачи ANN
- Выбраны подходящие под требования алгоритмы
- Найдены open-source библиотеки для реализации искомых алгоритмов
- Разработан поисковый сервис SimSearch
- Проведено сравнение производительности поисковых стратегий
- Разработан поиск похожих изображений с UI на основе SimSearch

Возможности дальнейшего развития

- Продолжение исследования задачи ANN
- Добавление новых поисковых стратегий и векторизаторов

Спасибо за внимание!