Проект
“Анализ цитируемости статей”

Выполнил: Лысенко Иван Егорович ММОВС23
Куратор: Илья Скворцов

Высшая Школа Экономики

Описание задачи

- **Цель работы**: предсказание текущих трендов в современных статьях по машинному обучению.
- **Задачи**:
  - [✓] Сбор данных и EDA
  - [✓] Анализ литературы по графовой кластеризации и NLP методам для кластеризации
  - [✓] Анализ графа цитирований с помощью простых методов графовой кластеризации (например, спектральной кластеризации)
  - [✓] Topic modeling
  - [✓] Кластеризация статей по контенту, например, используя эмбеддинги BERT [2]
  - [ ] Использование моделей, учитывающих как структуру графа, так и контент вершин (например, attributed graph clustering [6])
  - [✓] Написание Telegram бота

Описание данных

- **Исходная выборка**: arXiv dataset [5]
- **Преобразования над исходной выборкой**:
  - Выбирались только статьи 2023 года
  - Выбирались только статьи с arXiv тегами “cs.CV”, “cs.LG”, “cs.CL”, “cs.AI”, “cs.NE” и “stat.ML”
  - С помощью Semantic Search [3] API для каждой статьи загружалась дополнительная метаинформация, в частности ее ссылки
  - Для каждой ссылки также в список статей добавлялись ее ссылки с помощью информации из Semantic Search [3] API
- **Итоговая выборка**: граф цитирований, где каждая вершина – статья (название, абстракт, год, Semantic Scholar ID), а ребра – связи по цитированию.

EDA над графом

- Были невалидные вершины (например, с неверным Semantic Scholar ID). Они были удалены.
- 411988 вершин, из которых примерно 35000 – изначальные статьи 2023 года с arXiv
- 271 компонента связности, одна из которых размера 411706. Далее работа шла только с ней.
- Плотность большой компоненты связности $0.00011196709822370937 << 1$ – граф сильно разреженный

EDA над графом

Также можно почитать в репозитории тут:
https://github.com/taiypeo/mlds-project/blob/main/eda/EDA.md#loading-and-cleaning-the-graph
А также в ноутбуке тут:
https://www.kaggle.com/code/taiypeo/arxiv-pda

Эксперимент 1: спектральная кластеризация

- 30 собственных векторов Лапласиана большой компоненты связности, соответствующие 30 наименьшим собственным значениям
- k-means кластеризация с 8 кластерами. Размеры кластеров:

$$[654, 1, 1901, 3614, 1213, 3498, 1119, 23220]$$

- Облака слов по упрощенным с помощью nltk [1] текстам вида title + abstract, а также ручная проверка нескольких статей из каждого кластера не дают выделить темы кластеров

Итог: информации о связи вершин в графе недостаточно для адекватной кластеризации на темы.

Эксперимент 1: спектральная кластеризация

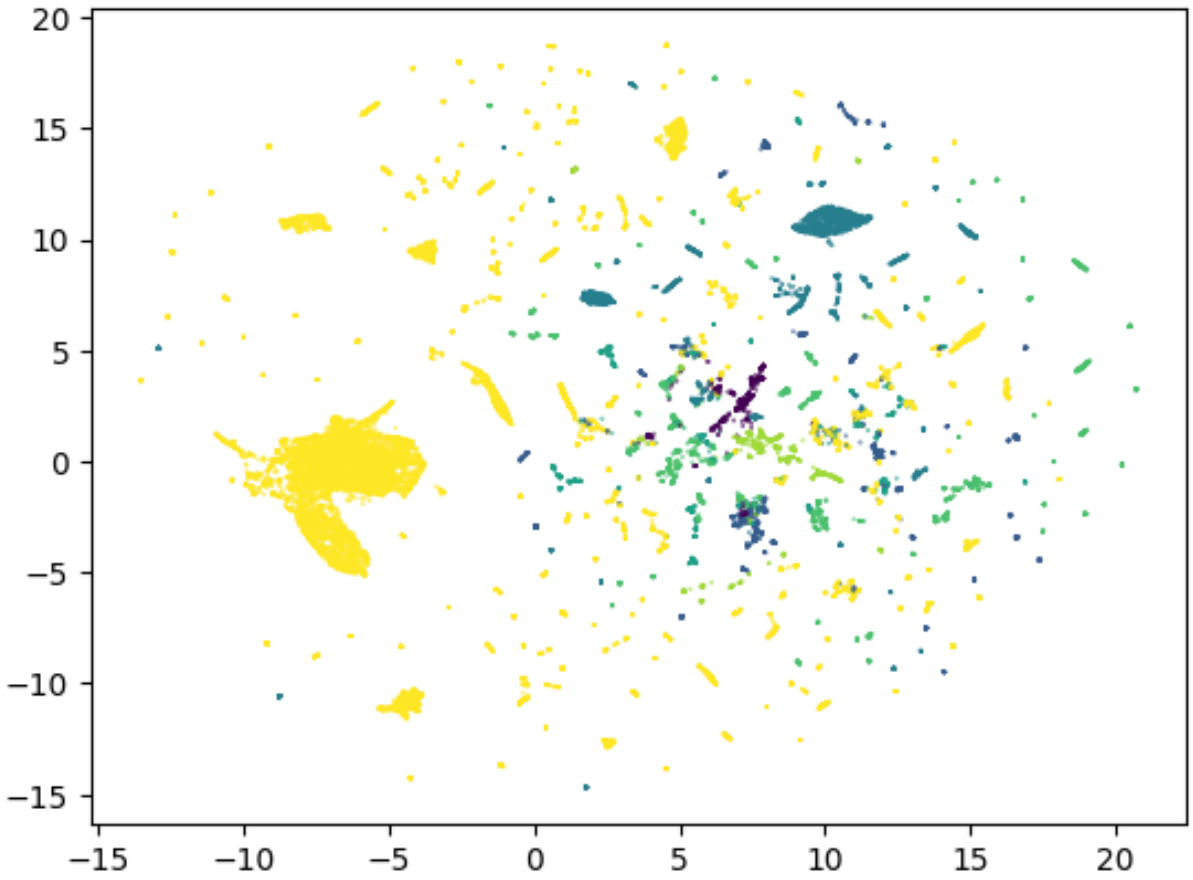

Рис.: UMAP [4] визуализация кластеров

Эксперимент 1: спектральная кластеризация

Также можно почитать в репозитории тут:
https://github.com/taiypeo/mlds-project/blob/main/eda/EDA.md#clustering-and-plotting
И также в ноутбуке тут: https://www.kaggle.com/code/taiypeo/spectral-clustering/notebook

Эксперимент 2: topic modeling, LDA

- Тексты вида title + abstract были упрощены с помощью nltk [1]
- С помощью gensim создался словарь токенов, встречающихся не менее, чем в 20 документах, но не более, чем в 50% документов
- Из словаря были убраны частые токены “algorithm”, “learning”, “training”, “data”
- На текстах с помощью словаря из 6715 токенов была обучена модель LDA с n_topics=10
- Итоговое распределение размеров тем:

$$[3367, 4206, 4100, 2094, 1932, 2182, 2830, 6291, 4969, 3492]$$

Итог: не все темы интерпретируемые, но есть довольно понятные.

Эксперимент 2: topic modeling, LDA

Также можно почитать в репозитории тут:
https://github.com/taiypeo/mlds-project/blob/main/clustering/README.md#lda
И также в ноутбуке тут:
https://www.kaggle.com/code/taiypeo/arxiv-lda/notebook

Эксперимент 2: topic modeling, HDP

- Препроцессинг такой же, как и у LDA
- Выбирать число тем вручную не нужно, большой плюс по сравнению с LDA
- Coherence scores гораздо ниже, чем у LDA

Итог: темы вышли очень плохие и непонятные.

Эксперимент 2: topic modeling, HDP

Также можно почитать в репозитории тут:
https://github.com/taiypeo/mlds-project/blob/main/clustering/README.md#hdp
И также в ноутбуке тут:
https://www.kaggle.com/code/taiypeo/arxiv-hdp

Эксперимент 3: кластеризация над эмбеддингами Sentence-BERT

- Препроцессинг: взял title и abstract у статей 2023 года, склеил через токен "[SEP]"
- Используемые предобученные модели Sentence-BERT:
  - allenai-specter - модель, специально предобученная на задачу semantic search между научными публикациями
  - all-mpnet-base-v2 - general-purpose Sentence-BERT модель, которая в среднем показывает себя лучше других моделей как на задаче semantic search, так и на других задачах, использующих текстовые эмбеддинги
- Над эмбеддингами проводил K-Means кластеризацию с 10 кластерами

Эксперимент 3: кластеризация над эмбеддингами Sentence-BERT

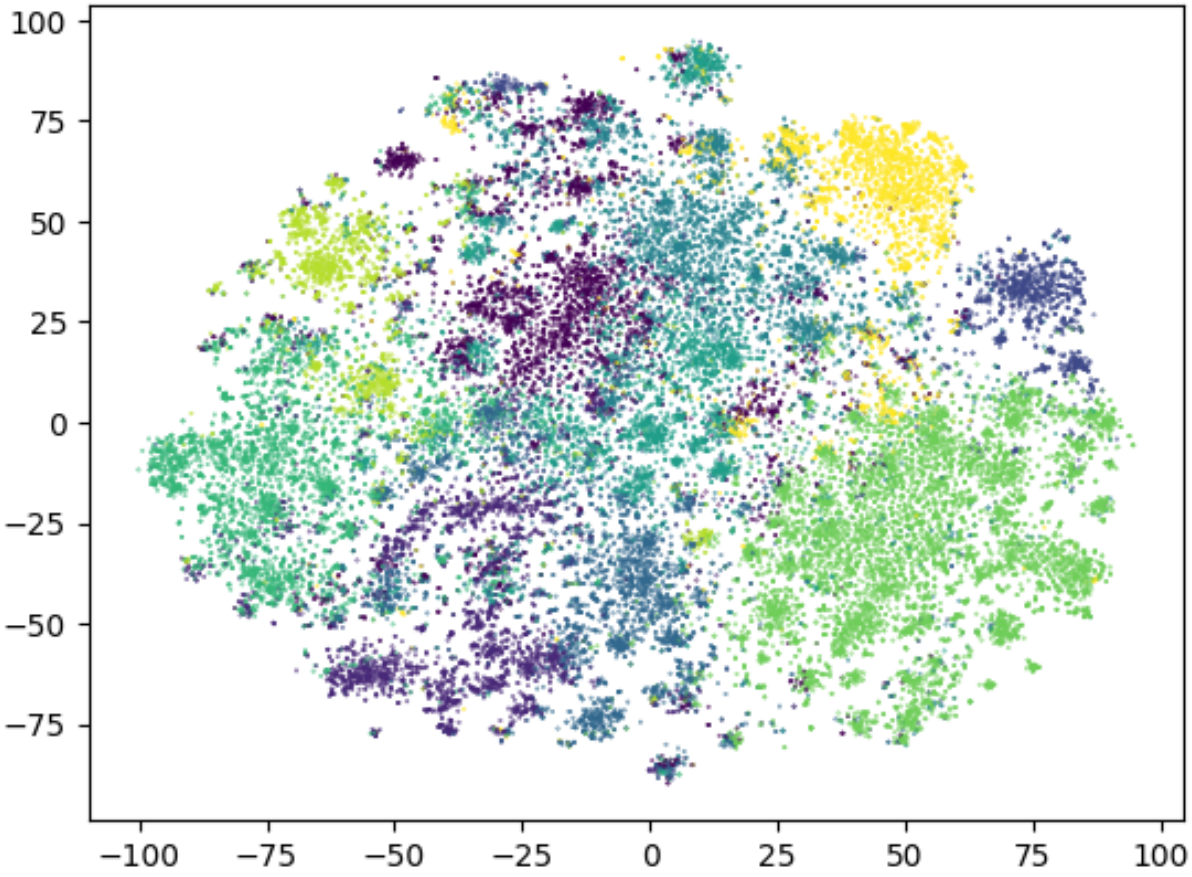

Рис.: t-SNE визуализация кластеров allenai-specter

Эксперимент 3: кластеризация над эмбеддингами Sentence-BERT

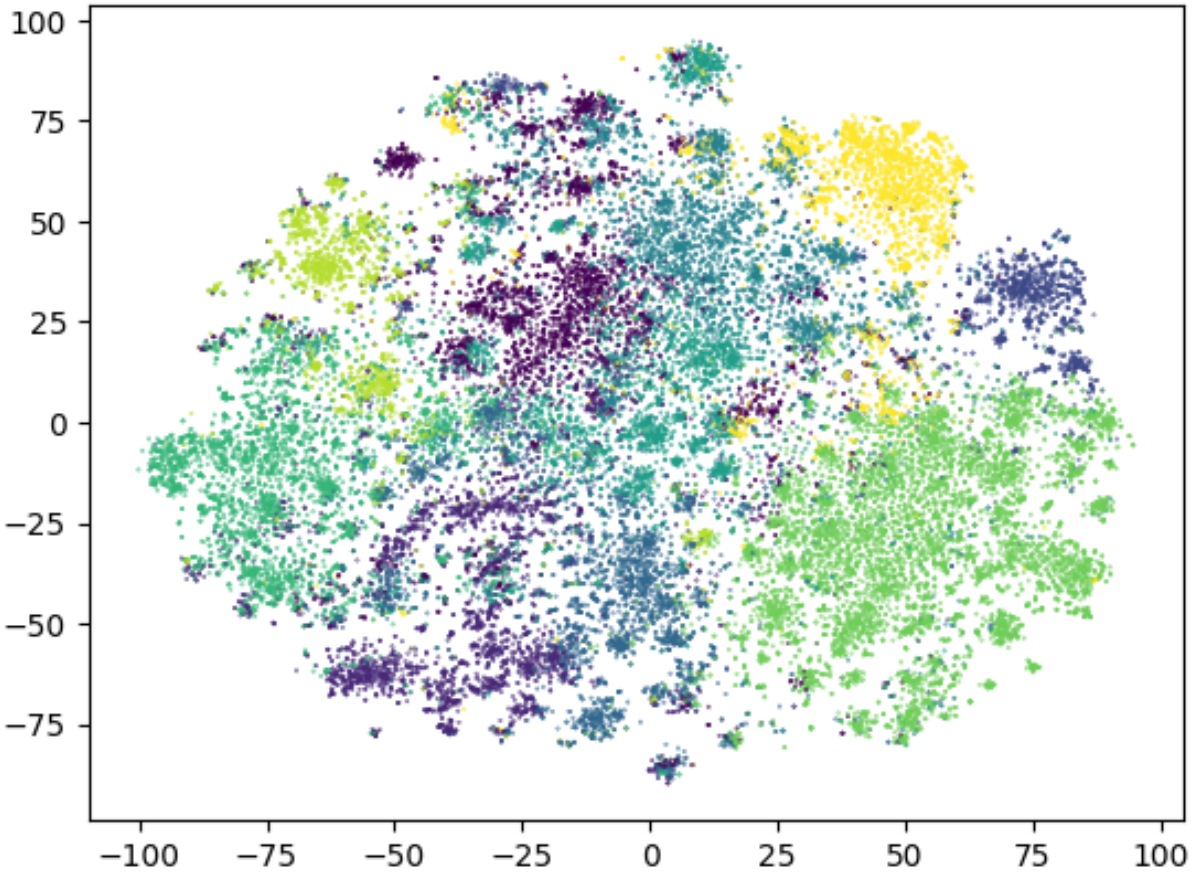

Рис.: t-SNE визуализация кластеров all-mpnet-base-v2

Эксперимент 3: кластеризация над эмбеддингами Sentence-BERT

- Видно, что пространства эмбеддингов этих двух моделей довольно плотно упакованы, и на глаз не выделяются никакие кластеры. Посмотрев на содержимое каждого кластера K-Means, можно понять, что для большинства из них не выделяется общая тема. В случае DBSCAN, меняя параметр eps, либо все примеры оказываются в кластере шума, либо (почти) все примеры оказываются в одном кластере.
- Вероятно, Sentence-BERT слишком много внимания обращает на нюансы в статьях, и из-за этого пригоден только для получения малого количества близких к данной статей, а не для задачи кластеризации всех статей на отдельные темы.

Итог: темы вышли очень плохие и непонятные.

Эксперимент 3: кластеризация над эмбеддингами Sentence-BERT

Также можно почитать в репозитории тут:
https://github.com/taiypeo/mlds-project/blob/main/clustering/README.md#sentence-bert--k-meansdbscan
И также в ноутбуке тут:
https://www.kaggle.com/code/taiypeo/arxiv-bert

Бот Telegram

`https://t.me/mlds_paper_analysis_bot`

- /start - старт бота
- /help - сообщение с описанием команд
- /get_random_papers - получить случайные статьи из кластера
- /get_paper_stats - получить статистику о статьях и кластерах
- /rate_bot - оценить бота
- /get_avg_rating - получить среднюю оценку бота

Бот Telegram: /get_paper_stats

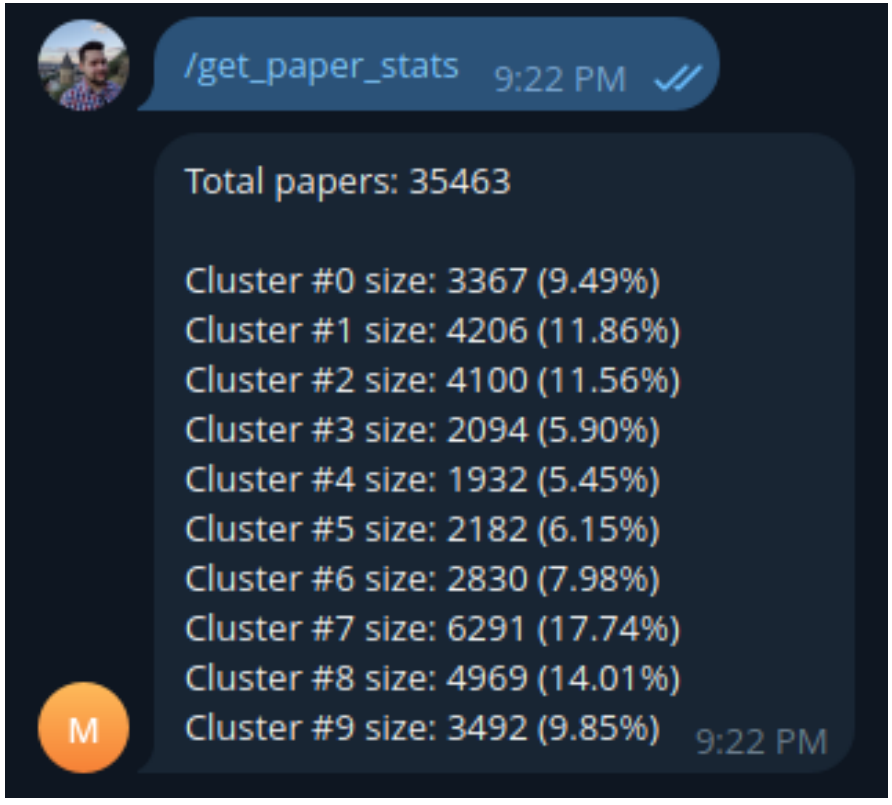

Рис.: Вывод команды /get_paper_stats

Бот Telegram: /get_random_papers

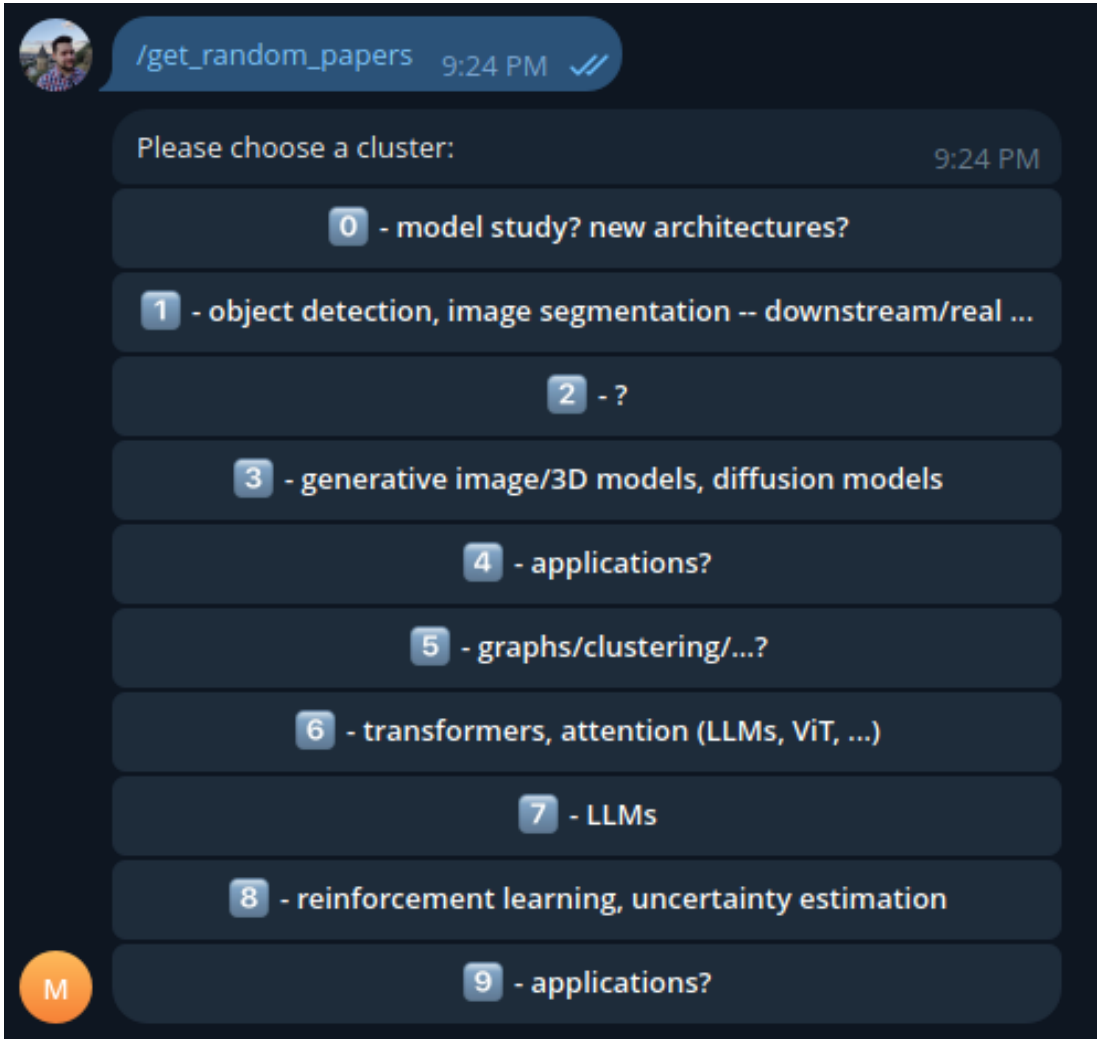

Рис.: Вывод команды /get_random_papers

Бот Telegram: /get_random_papers

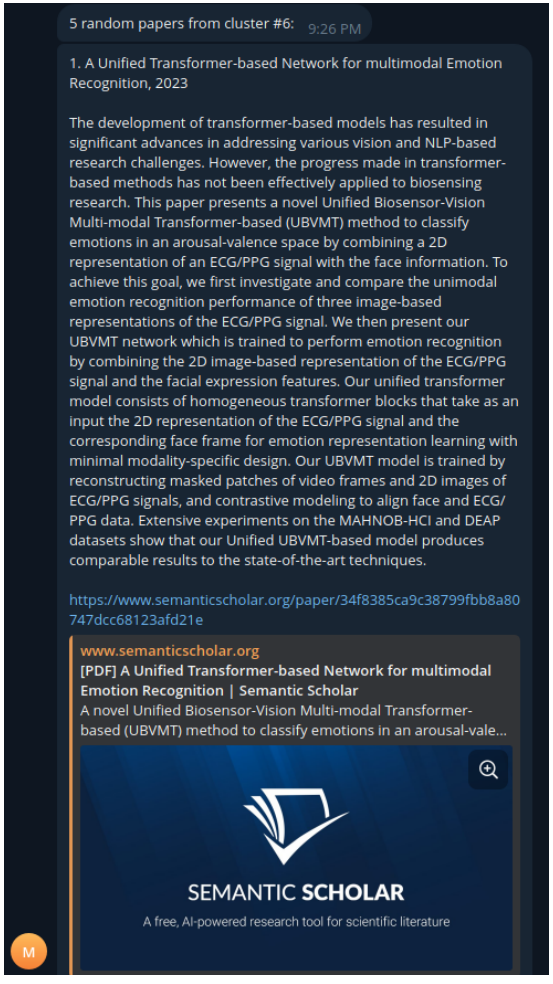

Рис.: Вывод команды /get_random_papers

Бот Telegram: бекенд сервис

Бот запускается через docker-compose, в качестве бекенда есть сервис на FastAPI. И бот, и бекенд обернуты в образы Docker и выложены на Docker Hub.

- GET /ping - тест живости сервиса
- GET /rating - получить среднюю оценку бота
- POST /rating?rating=<INTEGER> - оценить бот
- GET /random_papers?cluster=<INTEGER> - получить набор случайных статей из кластера
- GET /paper_stats - получить статистику о статьях и кластерах

References I

[1] Edward Loper Bird Steven. *Natural Language Processing with Python*. 2009.

[2] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[3] Rodney Michael Kinney et al. "The Semantic Scholar Open Data Platform". In: *ArXiv* abs/2301.10140 (2023). URL: https://api.semanticscholar.org/CorpusID:256194545.

[4] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020. arXiv: 1802.03426 [stat.ML].

References II

[5] arXiv.org submitters. *arXiv Dataset*. 2024. DOI: 10.34740/KAGGLE/DSV/7352739. URL: https://www.kaggle.com/dsv/7352739.

[6] Chun Wang et al. *Attributed Graph Clustering: A Deep Attentional Embedding Approach*. 2019. arXiv: 1906.06532 [cs.LG].