Data Clustering Contest: Round 1

> Перейти к английской версии документа

Задача конкурса – создание алгоритма ранжирования новостей.

Исходные данные

Участникам предлагается тестовый набор данных в формате HTML:

Архив 1, 01-07
Архив 2, 08-17
Архив 3, 18-21
Архив 4, 22-25

В ходе конкурса будут периодически публиковаться дополнительные наборы данных.

> Для проверки работ будет использоваться набор данных, отличный от тестовых. Он может содержать, в том числе, тексты с доменов, отсутствующих в тестовом наборе.

Задания конкурса

1. **Выделение текстов на русском и английском языках.** Алгоритм должен выделить все англо- и русскоязычные тексты, остальные языки не являются релевантными для данного этапа конкурса.
2. **Отделение новостей от других материалов.** Алгоритм должен оставить в списке только новости, отсеяв посторонние тексты, энциклопедические и справочные материалы и т.д.
3. **Группировка новостей по тематике.** Алгоритм должен распределить новости по следующим **7-ми** тематикам:

- Technology (включает Gadgets, Auto, Apps, Internet services)
- Sports (включает E-Sports)
- Entertainment (включает Movies, Music, Games, Books, Arts)
- Science (включает Health, Biology, Physics, Genetics)
- Other (новостные статьи, не попавшие в перечисленные выше категории)

4. **Группировка похожих новостей в сюжеты.** Алгоритм должен сгруппировать новости, написанные об одном событии/инфоповоде/сюжете, выбрав общий заголовок для группы новостей. Новости внутри сюжета должны быть отсортированы по релеватности.

5. **Ранжирование сюжетов.** Алгоритм должен сформировать списки сюжетов по тематикам, отсортированные по важности. Кроме того, нужно сформировать отсортированный по релеватности список сюжетов вне зависимости от тематики.

Прием работ

Работы будут приниматься в виде standalone приложения **tgnews** с CLI-интерфейсом. Приложение будет запускаться со следующими параметрами:

```
tgnews languages source_dir
tgnews news source_dir
tgnews categories source_dir
tgnews threads source_dir
tgnews top source_dir
```

где **source_dir** – путь до директории с HTML-файлами, содержащими тексты статей.

Требования к работам

Приложение не должно использовать сеть.
Приложение должно отличаться высокой относительной скоростью работы (по этой причине приложения, написанные на **C++**, могут получить преимущество).

запуском командой `sudo apt-get install ...`
Приложения будут тестироваться на серверах с Debian GNU/Linux 10.1 (buster), x86-64 с 8 ядрами и 16 GB RAM. Обязательно проверьте корректность работы программы на чистой системе перед отправкой работы.
Приложения, выполняющиеся дольше 60 секунд на каждые 1000 файлов, переданных в source_dir, рассматриваться не будут. Это ограничение должно выполняться для каждого из 5 запусков скрипта в процессе проверки работ.
На выходе требуется ZIP-файл (работы с размером архива, превышающим 200MB, с большой вероятностью будут оценены ниже) со следующей структурой.

```
submission.zip
  -> tgnews - запускаемый бинарный файл с интерфейсом, описанным ниже
  -> src - директория с исходным кодом приложения
  -> deb-packages.txt - текстовый файл с названиями пакетов внешних зависимостей, разделенных переводом ст
  -> * - дополнительные файлы, необходимые для работы программы (используйте относительные пути для доступ
```

Проверка работ

Проверка каждой работы будет происходить поэтапно. Приложение будет запускаться несколько раз с разными параметрами, описанными ниже.

> Работы будут проверяться на **англо-** и **русскоязычных текстах**.

1. Группировка по языкам

Запуск приложения:

```
tgnews languages source_dir
```

Результат должен быть выведен в формате JSON в STDOUT.

Формат ответа:

```
    "articles": [
      "981787246124324.html",
      "239748235923753.html",
      ...
    ]
  },
  {
    "lang_code": "ru",
    "articles": [
      "273612748127432.html",
      ...
    ]
  },
  ...
]
```

где:

lang_code – ISO 639-1 two-letter language code

articles – список имен файлов, содержащих тексты на языке **lang_code**

2. Отделение новостей от других материалов

Запуск приложения:

```
tgnews news source_dir
```

Результат должен быть выведен в формате JSON в STDOUT.

Формат ответа:

```
{
  "articles": [
    "981787246124324.html",
    ...
```

где:

articles – список имен файлов, содержащих новости

3. Группировка по тематике

Запуск приложения:

```
tgnews categories source_dir
```

Результат должен быть выведен в формате JSON в STDOUT.

Формат ответа:

```
[
  {
    "category": "society",
    "articles": [
      "981787246124324.html",
      ...
    ]
  },
  {
    "category": "sports",
    "articles": [
      "2348972396239813.html",
      ...
    ]
  },
  ...
]
```

где:

category – `"society"`, `"economy"`, `"technology"`, `"sports"`, `"entertainment"`, `"science"` или `"other"`

articles – список имен файлов, содержащих новости по тематике **category**

```
tgnews threads source_dir
```

Результат должен быть выведен в формате JSON в STDOUT.

Формат ответа:

```
[
  {
    "title": "Telegram announces Data Clustering Contest",
    "articles": [
      "6354183719539252.html",
      ...
    ]
  },
  {
    "title": "Apple reveals new AirPods Pro",
    "articles": [
      "9436743547232134.html",
      ...
    ]
  },
  ...
]
```

где:

title – общий заголовок для сгруппированного набора новостей
articles – список имен файлов, содержащих похожие новости, отсортированный по релеватности (релевантные выше)

5. Ранжирование сюжетов

Запуск приложения:

```
tgnews top source_dir
```

```
[
  {
    "category": "any",
    "threads": [
      {
        "title": "Telegram announces Data Clustering Contest",
        "category": "technology",
        "articles": [
          "6354183719539252.html",
          ...
        ]
      },
      {
        "title": "Apple reveals new AirPods Pro",
        "category": "technology",
        "articles": [
          "9436743547232134.html",
          ...
        ]
      },
      ...
    ]
  },
  {
    "category": "technology",
    "threads": [
      {
        "title": "Telegram announces Data Clustering Contest",
        "articles": [
          "6354183719539252.html",
          ...
        ]
      },
      ...
    ]
```

```
      "threads": [
        ...
      ]
    },
    ...
  ]
```

где:

category – `"society"`, `"economy"`, `"technology"`, `"sports"`, `"entertainment"`, `"science"` или `"other"`. Также обязательно должен присутствовать блок с `category="any"`, содержащий отсортированный по важности (важные выше) список сюжетов вне зависимости от тематики.
threads – список сюжетов, отсортированный по важности (важные выше). Каждый сюжет содержит название и список статей. Для блока `category="any"` также содержит поле **category**.
title – общий заголовок для сюжета.
articles – список имен файлов статей по этому сюжету, отсортированных по релеватности (релеватные выше).

Пояснения

Рекомендуемый размер итогового архива – меньше 200 MB. Вы можете отправить файл до 1,5 GB, но такие работы с большой вероятностью будут оценены ниже.
Ограничение на 60 секунд / 1000 файлов должно выполняться для каждого из 5 запусков скрипта в процессе проверки работ.