

004.021

Орлов А. С., Севрюков С. Ю.

Реализация алгоритма поиска по субтитрам YouTube

1. Введение. В наши дни платформа хранения, доставки и показа видео YouTube пользуется огромной популярностью. Она обладает большим спектром возможностей, что позволяет упрощать процессы работы с видеоматериалами. Одним из таких процессов является поиск по тексту субтитров YouTube, генерируемых автоматически, что предоставляет возможность реализации алгоритма автоматизации этого процесса. Это позволит повысить производительность организаций, активно использующих YouTube в работе.

Примером такой организации может быть центр развития электронных образовательных ресурсов Санкт-Петербургского государственного университета, который занимается созданием и выпуском онлайн-курсов. В процессе их создания видеолекции просматриваются авторами для поиска ошибок. В курсах для публикации на платформе Coursera в среднем 4-6 модулей; на платформе Открытое Образование — 10-12 модулей. Каждый модуль включает в себя около 1,5 часа видеолекций. На один курс приходится минимум 6 часов итогового видеоматериала, не считая чернового. Для упрощения ориентации в таком массиве данных было решено разработать описываемый алгоритм.

2. Постановка задачи. В процессе решения задачи и реализации алгоритма были сформулированы следующие начальные условия:

1. Массив видео, опубликованных на YouTube, по которым будет производиться поиск.
2. Данные учётной записи Google.
3. Наличие речи в видео.

Орлов Антон Сергеевич – студент, Санкт-Петербургский государственный университет; e-mail: ion6431@gmail.com, тел.: +7(999)229-50-25

Севрюков Сергей Юрьевич – старший преподаватель, Санкт-Петербургский государственный университет; e-mail: s.sevryukov@spbu.ru, тел.: +7(911)910-94-88

Необходимо подчеркнуть, что качество генерируемых субтитров напрямую зависит от качества звука в видео [?]. Описываемая реализация направлена на повышение эффективности работы с видео в формате, предустановленном ЦРЭОР, однако может быть применена и к любым другим видео, удовлетворяющих начальным условиям.

Цель заключается в реализации алгоритма, на вход которому подаётся несколько видео, в которых присутствует речь, данные учётной записи Google, с которой были загружены видео, и поисковый запрос. Результатом является список ссылок на видео, отсортированных в порядке уменьшения релевантности.

3. Получение текста видео. Извлечение текста будет производиться с помощью YouTube Data API. В него входит метод `captions.download`. Согласно документации, метод требует авторизации, следовательно, её необходимо реализовать.

3.1. Авторизация. YouTube API поддерживает два способа авторизации: OAuth 2.0 и API Key. Второй способ не предусматривает получение прав, поэтому в дальнейшем он рассматриваться не будет. Использование метода запрашивает наличие хотя бы одного из двух разрешений: `youtubepartner` или `youtube.force-ssl`. Они указываются в параметре `scope`. Оставшиеся действия описаны в руководстве API [?]. После прохождения авторизации становится возможным использование метода получения субтитров.

3.2. Получение субтитров. Метод `captions.download` принимает в качестве обязательного параметра `id` дорожки субтитров. Её можно получить с помощью метода `captions.list`. Google накладывает условие, согласно которому субтитры возможно получить только из видео, загруженных пользователем. Данное условие может быть снято, если пользователь разрешает взаимодействие третьим лицам с его видео. Организации, которым требуется поиск по собственным видео, это условие не создаст препятствий.

4. Индексация. Рассмотрим процесс создания обратного индекса. Сначала необходимо каждое слово привести к начальной форме. Далее слову сопоставляется список видео, и список временных промежутков, в которых это слово встречается.

Полученный текст имеет следующий формат: в первой строке находится указание временного промежутка, во второй — текст, который соответствует этому промежутку, и пустая третья строка. Все дальнейшие строки подчиняются этому правилу. Это позволяет со-

здать цикл с шагом 3, внутри которого будет происходить наполнение индекса. Первая строка обрабатывается с помощью регулярного выражения: $(\sim|\backslash n)(\backslash d\{1,2\}):(\backslash d\{2\}):(\backslash d\{2\})\backslash.(\backslash d\{3\})$. С помощью этого можно получить значение секунд, которое соответствует началу временного промежутка видео с нужным словом. Вторая строка делится пробелами на слова, каждое из которых приводится к нормальной форме, записывается как ключ индекса, а к значению добавляется видео и соответствующие секунды.

5. Реализация метода поиска. Для того, чтобы найти видео, которое удовлетворяет поисковому запросу, необходимо вычислить меру схожести запроса и всех видео, в которых присутствуют слова из запроса. В качестве такой меры можно выбрать cosine similarity. Подробный разбор алгоритма описан в книге Introduction to Information Retrieval [?]. Результатом работы метода является список видео с сопоставленными им cosine similarity. Искомый результат будет получен при сортировке по убыванию.

6. Использование Elasticsearch. Рассмотрим реализацию поиска с помощью Elasticsearch. Чтобы проиндексировать субтитры, полученные с помощью вышеизложенных методов, необходимо выделить текст без отметок о времени. Они находятся в каждой третьей строке, начиная с первой. Также каждая третья строка является пустой, поэтому их можно игнорировать. Для повышения качества поиска возможно индексировать каждую строку отдельно, чтобы пользователь получал не список видео, а список фрагментов, в которых употребляются слова из его запроса. Это может быть полезно для поиска конкретной темы, которая обсуждалась в видео. Elasticsearch является RESTful сервисом, из-за чего для каждой строки необходимо делать два запроса. Это сильно понижает производительность системы.

7. Метод деления субтитров на фрагменты. Субтитры состоят из фраз с привязкой к временному промежутку в видео. Временные промежутки могут пересекаться (тогда во время просмотра этого видео с субтитрами пользователь видит сразу несколько строчек субтитров). Однако если существуют моменты видео, в которых нет речи, в этом временном промежутке не генерируются субтитры. Следовательно, когда делаются паузы в речи, субтитры тоже не генерируются. Такие паузы можно вычислить, сравнивая конец

временного промежутка с началом следующего. Если второй начинается раньше, чем кончается первый, то есть пересечение, и паузы в речи не было. В противном случае пауза была, и это может значить, что сменилась тема разговора. По таким паузам происходит деление файла субтитров на документы. Каждый из них индексируется, и поиск будет идти уже по фрагментам видео.

8. Тесты. Ниже приведена информация о затратах времени для индексации с помощью рассмотренных методов. Все видео являются лекциями, поэтому количество моментов, в которых нет речи, минимально.

Таблица 1. Сравнение

Длительность видео	Inverted Indexing	ElasticSearch
7:52	656мс.	17,9с.
1:58:35	1358мс.	3м. 6с.
2:55:02	1287мс.	5м. 4с.
35:34	919мс.	2м. 2с.
53:19	989мс.	3м. 46с.

Таким образом, ElasticSearch имеет гораздо более низкую скорость работы, но описываемый алгоритм имеет приложения в таких ситуациях, когда скорость не влияет на продуктивность системы и компании-заказчику выгоднее работать с готовыми решениями.

Модификация сегментации субтитров позволяет как улучшить, так и ухудшить время индексации. Это зависит от формата видео, по которым настраивается поиск. Например, для ЦРЭОР эта модификация может улучшить качество алгоритма, так как во время съёмок авторы периодически делают паузы, чтобы обозначить конец модуля, темы и т.п.

9. Вывод. Для повышения эффективности данный результат может быть интегрирован в информационную систему, в которой присутствуют компоненты взаимодействия с материалами, опубликованными на YouTube. Рассмотренные вариации методов поиска показывают, что они могут варьироваться в зависимости от формата видео, для которых необходимо внедрение поиска. Алгоритм авторизации может быть модифицирован таким образом, чтобы пользователю не нужно было проходить авторизацию самостоятельно.

Литература

1. Manning C. D., Raghavan P., Schutze H. Introduction to Information Retrieval. 2008. 151 p.
2. Как создать субтитры автоматически — Справка — YouTube [Электронный ресурс]: URL:<https://support.google.com/youtube/answer/6373554?hl=ru> (дата обращения: 14.03.18).
3. Implementing OAuth 2.0 Authorization | YouTube API | Google Developers [Электронный ресурс]: URL:<https://developers.google.com/youtube/v3/guides/authentication> (дата обращения 14.03.18)