YOUStatAnalyzer: a Tool for Analysing the Dynamics of YouTube Content Popularity

Mattia Zeni, Daniele Miorandi, and Francesco De Pellegrini
CREATE-NET
via alla Cascata 56/D, 38100
Trento, Italy
email: name.surname@create-net.org*

ABSTRACT

Understanding the dynamics of on-line content popularity is an active research field with application in sectors as diverse as media advertising, content replication and caching and on-line marketing. In most cases, scientists have focused on user-generated contents, which are freely accessible through different on-line services. Among such services, the incumbent one is indeed YouTube. This online platform was launched in 2005 and it currently features more than 6 billions hours of video watched every month (almost one hour per person on Earth), with more than 100 hours of videos uploaded every minute and 1 billion unique users per month[1]. In order to analyze or predict content popularity, statistics about viewers, watch time and shares must be retrieved. The YouTube APIs, however, do not allow third parties to retrieve such an information in an open and accessible way. In order to overcome this problem, we have developed a framework, based on Web scraping techniques and big data tools, for the collection and analysis of YouTube video content popularity at scale. Our framework, called YOUStatAnalyzer, enables researchers to create their own dataset, according to a number of different search criteria and analyse them to extract relevant features and significant statistics.

Categories and Subject Descriptors

C.4 [**Computer-Communication Networks**]: Performance of Systems—*Measurement techniques*; H.3.5 [**Information Storage and Retrieval**]: Online Information Service—*Web-based services*

General Terms

Measurement

This work has been partially supported by the European Commission within the framework of the CONGAS project FP7-ICT-2011-8-317672, see www.congas-project.eu 1http://www.youtube.com/t/press statistics/

VALUETOOLS 2013, December 10-12, Torino, Italy

DOI 10.4108/icst.valuetools.2013.254391

Keywords

YouTube, Big Data, Data Collection, Analyzer

1. INTRODUCTION

Understanding how digital contents are accessed, diffused and shared through online platforms is a key challenge for a number of application, including the design of effective replication/caching schemes for content distribution networks, the design of online marketing campaings, etc. In this currently very active research field the tools employed are largely represented by statistical and artificial intelligence algorithms applied to massive datasets. The aim is to analyse and model the popularity of online contents, and to capture the role played by social relations onto the dynamical processeses governing popularity.

Research in the field relies on the ability of retrieving data on the actual dynamics of content popularity on online platforms. In practice, YouTube exposes APIs that allow retrieving *some* data on videos. Yet, these data (the one accessible through the `Data` API) include only the *cumulative viewcount*, i.e., the total number of viewes a content has experienced. Diverse statistics exist though, spanning the number of users who accessed the content, how much time spent on a content page, the number of subscribers and shares generated. However, those metrics are made available to the content owners only. More exactly, to the channel managers owning the channel publishing the content. This represents a strong limitation, banning most researchers from the opportunity of openly accessing detailed data on popularity dynamics.

However, some information still displays in the videos' webpage in the form of graphs. Those graphs show diverse popularity parameters such as views, subscribers, watch-time and shares in a daily and cumulative manner. The resolution there is one day: researchers have then developed Web scraping techniques, by which they managed to build databases of YouTube content statistics. The primary example of such an approach is [5]. This approach has been based on a vulnerability in the way such graphs were generated. In practice, the browser was retrieving such data from the YouTube servers. Servers sent them in clear, whereas browsers would use them to generate the graphs to be displayed.

But, the situation changed in early June 2013, when YouTube modified the way how such data was accessed and processed. A single-use session token was inserted in order to ensure that data about statistics could not be automatically re-

trieved. This change, leaving aside technicalities, had the effect of making the solutions proposed in the past useless. A new approach was needed to provide the research community with an open way to retrieve and analyze data on the popularity of user generated contents.

In this paper we introduce YOUStatAnalyzer, a tool for researchers to build in a fast, flexible and reliable way a database populated with detailed statistical data on the popularity of YouTube user generated contents. The YOUStatAnalyzer builds upon Web scraping techniques, powered by the usage of a proxy, and NO-SQL database technologies. The YOUStatAnalyzer supports various search criteria, including video IDs, keywords, categories, YouTube standard feeds (e.g., most popular) and random keywords. YOUStatAnalyzer can be used to build large databases, to be analysed and studied by researchers using advanced data science techniques.

Since in the past few years various relevant works have been published, we review some of them here. The closest work to ours is [5]: Web scraping techniques were used in order to build a database with YouTube statistics. The criteria used to select videos to be analyzed are: (1) YouTube top lists, (2) random videos, searched through the API and starting from random keywords and (3) selection based on the YouTomb[2] dataset, containing videos deleted from YouTube due to copyright violations. As discussed above, the approach used in [5] is not applicable any longer due to the changes performed in the YouTube backend as of June 2013. In [3] the YouTube Data APIs was levered to populate a database with videos meta-data information, which was then processed to obtain significant statistics. In [2] video meta-data, obtained through YouTube data API, was used as well. In [4] a combination of viewcount information retrieved from the YouTube data API with additional information crawled from the video's page is used. In [1] the same process is used to retrieve statistics. No further details are given about the criteria used to determine which videos to analyze.
The remainder of this paper is organized as follows. Sec. 2 provides a general overview of YouTube statistics, in particular how statistical data about videos are structured and presented. Sec. 3 details the YOUStatAnalyzer technological infrastructure. A use case, showing how YOUStatAnalyzer can be effectively leveraged by researchers and practitioners for analysis purposes, is presented in Sec. 4. Sec. 5 concludes the paper and outlines some directions for future work.

2. YOUTUBE VIDEO STATISTICS REPRESENTATION

YouTube maintains a number of statistics on video access: those can be viewed on the specific videos' pages, unless the content states otherwise. The publisher of a content can decide whether statistics should be accessible to viewers or not, and this happens for a wide range of contents, e.g., sport events or several commercial-related videos. Such statistics are logged according to the following fields:
`watch-time`: it indicates the time spent in minutes by viewers watching the video. This particular field was introduced in Oct. 2012. Such a metric is not made available for videos published before such date;
`subscribers`: the number of subscription to the relevant YouTube channel generated by the specific video;
`views`: the number of views scored by the video, i.e., the number of YouTube users who *started* watching the video;
`shares`: the number of times the video was shared by YouTube users;
`day`: the time (in days) at which the four aforementioned statistics were sampled (used for plotting purpose).

When disclosed, all such fields are provided both on a cumulative (i.e., since the publication of the content) and on a per-day basis.

Also, it is worth noting that until early June 2013 data were provided only in terms of `views` and limited to 100 samples. In the case of a video of lifetime larger than 100 days, data was undersampled. For videos with less than 100 days, daily sampled data were provided. In our data scraping process we have been able to measure up to 2352 samples for a video.

Our solution let retrieve the public statistics shown in the videos webpage in combination with some useful meta-data information. A complete list of fields we can collect for each video is shown through an example in Table 1.

_ID	0ao1UwIYQCg
Caterogy	Howto
Description	Older project, new uplo...
Title	Make A Disk Sander
Author	John Heisz
Published Date	2012-07-08T00:09:58.000Z
Access Control	['comment': ['allowed'], ...]
Comments #	50
Related Videos	[4kU2veNhyKI,A4PzyaoRYk0...]
Duration	207
Video Type	video/3gpp
Views daily/cumulative	[528, 296, 206, 182, 114, 83, 126...] [824, 1030, 1212, 1492, 1326, 1409...]
Subscribers daily/cumulative	[1, 0, -1, 1, 0, 0, 2, 3, 0, 1, 0, 1, 0, 2...] [1, 1, 0, 1, 1, 1, 3, 6, 6, 7, 7, 8, 8...]
Shares daily/cumulative	[0, 0, 0..., 2, 0, 0, 0, 0, 0, 0, 0...] [0, 0, 0..., 2, 2, 2, 2, 2, 2, 2...]
Day	**from:** Sun, 08 Jul 2012 00:00:00 GMT **to:** Sat, 07 Sep 2013 00:00:00 GMT

Table 1: Collected parameters of a YouTube video.

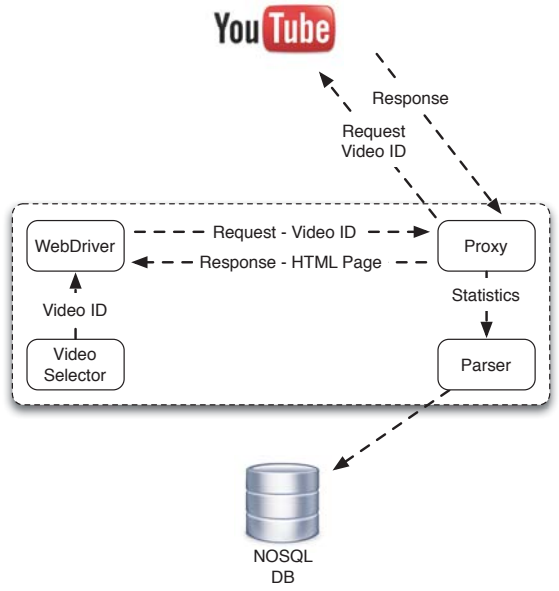
3. YOUStatAnalyzer: PLATFORM DETAILS

The overall architecture of YOUStatAnalyzer is depicted in Fig. 1a. The operations performed by the script follow four steps:
1. **Configuration**: the first step is to establish which videos should be analysed. YOUStatAnalyzer currently supports four types of configuration:

- direct method: a list of video IDs can be specified;
- category keyword-based: the user specifies a set of keywords and categories and the tool searches within all videos matching the given category and the chosen keywords;
- YouTube "standard feeds". YouTube supports a set of standard feeds [3], which is a way to find `most_popular videos;`

[2]http://youtomb.mit.edu/

[3]https://developers.google.com/youtube/2.0/reference



(a) Architecture of YOUStatAnalyzer.

```
[
        "_id" : "",
        "accessControl" : {
           "comment" : {
              "permission" : "allowed"},
           "list" : {
             "permission" : "moderated" }}
                ,
        "category" : "Music",
        ...,
        "views" : {
           "cumulative" : {
              "data" : [100.0, 543.0]},
           "daily" : {
             "data" : [643.0, 786.0]},
        "shares" : {
           "cumulative" : {
              "data" : [9.0, 16.0]},
           "daily" : {
              "data" : [1.0, 8.0, 7.0]}},
        "day" : [1342828800.0, 1342915200.
           0]
        }
],
```

(b) YOUStatAnalyzer MongoDB structure.

Figure 1

- `random keyword-based: search using random words from the WordNet dictionary` [4].

The user can define one of the four aforementioned configurations setup or a combination thereof. The Video Selector component, as represented in Fig. 1a is responsible for calling YouTube servers using the YouTube Data API and extracting a list of video IDs to be analysed.

2. **Browser Automation**: once the application has collected the ID of the videos to be analysed, it needs to extract the statistics of each one. To do that, we need a framework to automate the browser operations, simulating the user interaction with the page and the click on the "Statistics" button. To the best of our knowledge, this is the only way to download the statistics data, since from early June 2013 YouTube infrastructure changed, blocking direct calls to links of the type: `http://www.youtube.com/ insight_ajax? action_get_statistics_and_data=1&v=...` using `curl` or similar tools. YOUStatAnalyzer uses WebDriver Selenium for Python[5] in order to simulate the Firebox browser. The WebDriver was configured in such a way to redirect all HTTP/HTTPS traffic through our proxy on "**localhost:8080**", as explained next.

3. **Proxy**: we used a proxy to capture the YouTube AJAX response to the HTTP POST request for video statistics generated by the WebDriver. In fact, clicking on the "Statistics" button calls a Javascript whose response is an `xml` file sent to the browser. Filtering HTML header, we can distinguish the packets containing statistics data from which we can extract the compressed (using `gzip`) body.

4. **Data pre-processing**: once the proxy has captured the response containing the statistics, some processing is needed in order to extract relevant data. First, the message body gets unzipped. Second, the content is parsed using regular expressions to isolate the useful data. The result is then loaded into a NOSQL database (in our case: MongoDB), which can be used for running analytics. Every video is stored as a single record into the database, and all the additional information about it are saved as pairs of key-value in JSON format. The record structure is depicted in Fig. 1b.

The tool has been written in `python`, given the wide availability of useful libraries and the good match with big data analytics applications. The main libraries we used are: `re`[6], `webdriver`[7], `pymongo`[8], `urllib2`[9] and `xml`[10].

4. USING YOUStatAnalyzer

The software can be configured through an external .xml configuration file. The user can set parameters about the database, including the IP address of the server to connect to and the listening port of the mongoDB instance (default: 27017). This allows to run a measurement campaign on multiple devices, while putting all the results in one single database. Through the xml configuration file, the user can also set analysis parameters, by setting the *analysis mode* to active and by detailing the *campaign* type, i.e., the criteria according to which the videos to be analysed are selected. A campaign can include a number of different criteria according to which files are chosen, as explained in the previous section. Example of criteria supported are: `orderBy`, `number_of_results`, `video_duration`, `video_category`, `hd`,

[4] WordNet® is a large (more than 115k entities) lexical database of English, `http://wordnet.princeton.edu/wordnet/`

[5] http://docs.seleniumhq.org/docs/

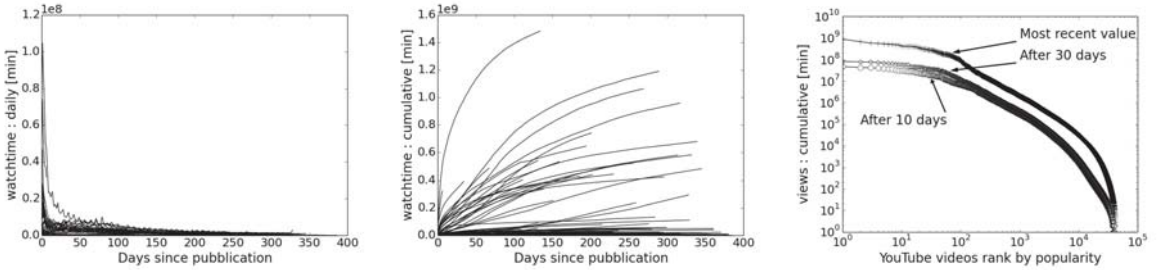
[6] http://docs.python.org/2/library/re.html

[7] http://docs.seleniumhq.org/docs/

[8] http://api.mongodb.org/python/current/

[9] http://docs.python.org/2/library/urllib2.html

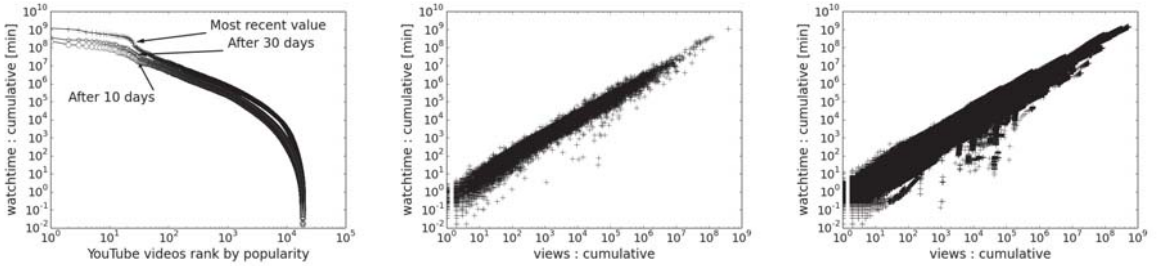
[10] http://docs.python.org/2/library/xml.dom.minidom.html



(a) Watchtime: daily trend for videos in YouTube Music category.

(b) Watchtime: cumulative trend for videos in YouTube "music" category.

(c) Views: cumulative trend versus videos ranking.



(d) Watchtime: cumulative trend versus videos ranking.

(e) Watchtime: cumulative versus videos views referred to the 50th day since publication.

(f) Watchtime: cumulative versus videos views referred to all days.

Figure 2: Sample analysis of videos belonging to YouTube 'music' category.

`safeSearch`, `time`, `feed`, `region`, etc.

Once the user has set these parameters, YOUStatAnalyzer can be executed from the command line. Results can be plotted using standard tools extracting data directly from the database.

In terms of scalability, each video analysed uses approximately 35kB of disk space in the database. The time needed to analyse one video ranges from 5s (fiber link connection, desktop pc) to 15s (ADSL connection, standard laptop).

We report hereafter some examples of plots obtained through our data collection and analysis tool. We selected the watchtime trend for videos belonging to YouTube "music" category for a total of more than 2000 videos. The result can be seen in Fig. 2a (daily) and 2b (cumulative). Both curves describe how new contents initially raise a large spike of views and then degrade progressively over time. In Fig. 2c and 2d the trend of the views and of the watchtime are shown versus the videos ranking. We note a plateau with high values both for the view and the watchtime for the 100 most popular videos. Also, in Fig. 2e and 2f we plot an interesting analysis of the watchtime against the number of views, for the 50th day since publication and for the whole lifetime. It is interesting to notice that, while the watchtime is expected to be sublinear compared to the views, the relation becomes more and more linear for higher number of views. This corresponds to the intuition: we expect that the most popular contents are watched entirely by almost all viewers who access them.

5. CONCLUSIONS

We designed and implemented an innovative approach for retrieving popularity statistics from YouTube. YOUStatAnalyzer allows researchers to quickly build large datasets for analysing the evolution and dynamics of content popularity. The tool is flexible and provides support for different types of queries. Next steps include the development of a web-based interface for visualising the statistics and the release — as open data — of a large database to the research community.

6. REFERENCES

[1] Altman, E., De Pellegrini, F., El Azouzi, R., Miorandi, D., and Jiménez, T. Emergence of equilibria from individual strategies in online content diffusion. In *Proc. of IEEE INFOCOM NetSciCOmm* (Turin, Italy, April 14-19 2013).

[2] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., and Moon, S. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proc. ACM SIGCOMM IMC* (New York, NY, USA, 2007), ACM, pp. 1–14.

[3] Chatzopoulou, G., Sheng, C., and Faloutsos, M. A first step towards understanding popularity in YouTube. In *Proc. of IEEE INFOCOM* (2010), pp. 1–6.

[4] Cheng, X., Dale, C., and Liu, J. Statistics and social network of YouTube videos. In *Proc. of IWQoS 2008* (2008), pp. 229–238.

[5] Figueiredo, F., Benevenuto, F., and Almeida, J. M. The tube over time: characterizing popularity growth of YouTube videos. In *Proc. of the ACM international conference on Web search and data mining* (2011), ACM, pp. 745–754.