Real-time Subtitle Rating

Guillermo Guridi and Aref Moradi

KTH University, Data-Intensive Computing

1 Introduction

In the current digital age more and more people have access to the internet and through it to various media. One of the main audiences of this type of media are the youth. We propose a project that helps create safer environment for younger people in their interaction with movies and TV-series and might help parents be able to better monitor which content their children are exposed too.

In this project we build a real-time subtitle rating system which uses streaming and text processing tools to provide an automatic rating of movies and TV-series by using the subtitle data from such sources. The main purpose of the project is to show how these tools can be leveraged to create a simple yet powerful system to provide this functionality. Also, alongside the system a simple model will be provided.

2 Data Gathering

This project is separated in two independent parts. The first part is responsible for finding and gathering the required subtitle data for the movies and TV-series as well as their age ratings and training a classification model using this data.

All of the data is gathered in an automated fashion and only a list of movie names is needed as input. Two main API's are used for this phase. One API is used for extracting the movie's age rating in Sweden [1] and the other API is used for extracting the English subtitle of the same movie [3].

Connecting to the API and downloading the data is done in parallel by leveraging Spark as can be seen in the Training notebook. The beginning list of the movie names is parallelized and used to download both the subtitle and rating data of the movies.

After receiving the raw data from each of these API's some processing is done on the data using Spark to prepare the data in a format that is ready to be used as training data in the classifier.

3 Data Processing

As mentioned in the previous section Spark is used to transform the raw data gathered from the API's to a proper format for training the classification.

The subtitle data is in srt format which contains timestamps for the subtitles which are not needed for the training of the model. These files are transformed into plain text using a python library for manipulating subtitles [2]. The resulting plain text is then divided into smaller texts to be used in training the classifier.

The raw data extracted from the IMDB database contains all kind of information about the movie which is not needed in this project. Hence, in this part the rating data for Sweden is extracted using Spark maps and some minor text processing.

The resulting data from the previous steps is ready to be used to train the model.

4 Model Training

In this part we use the subtitle data from the previous section as training data and the ratings as the label for each of the pieces of text. The text are transformed into vectors using the *scikit HashingVectorizer* in the data processing section and used in the training phase. The resulting model is saved using the *Pickle* library to be used in the second part of the project which will use the model and Kafka to provide an age rating for subtitle content in real-time.

5 Real-time processing

The real-time processing and prediction of the rating is done in this section. This part of the project consists of the three notebooks processor, receiver and sender. The sender notebook reads subtitles from an example subtitle file and emits them on Kafka. The processor receives the subtitles in fragments and predicts their Swedish age rating using the model trained in the previous sections. After this step the rating is emitted again on Kafka where the receiver consumes them. The current receiver simply prints the age rating.

6 Implementation notes

The project has been implemented using Jupyter and Docker. The project can be simply started by running the 'docker-compose' file and then opening the Jupyter notebook in the browser. The project has been completely dockerized. Also, as mentioned in the previous sections the training section of the project isn't the main purpose of the project and the model can be provided from any other external source as long as it is saved with the *Pickle* format. Although, the current setup mines the training data in a parallelized fashion and saves the model automatically so the process of updating the model can also be done automatically.

References

1. imdb python library for age rating kernel description. https://pypi.org/project/IMDbPY/
2. Subtitles manipulation python library. https://pypi.org/project/pysrt/
3. Subtitles python library. https://pypi.org/project/subliminal/