Extracting Conference Highlights Using Machine Learning Systems

Hasan Bank
TU Berlin
Berlin, Germany
hasanbank@gmail.com

Christian Fuhrhop
Supervisor of the Project
Fraunhofer FOKUS
Berlin, Germany
christian.fuhrhop@fokus.fraunhofer.de

***Abstract*—This project aims to summarize the conference talks as cutting a section from the related videos. In order to achieve this purpose, a predefined machine learning API is used.**

***Index Terms*—Conference talk, Microsoft Face API, Emotion analysis**

I. Introduction

This paper describes the project that is done in the scope of Advanced Web Technologies Project during winter semester 2018 in TU Berlin supervised by Christian Fuhrhop, Fraunhofer Fokus. Source code and presentations are available at GitHub repository [6].

Fraunhofer Fokus organizes a conference annually that includes several speakers and long talks. The main idea of this project is whether a summary or trial can be built automatically or not. If the talks are summarized automatically, they can be combined and obtained a trial video that seems created by the human force.

II. Similar Projects and Related Technologies

In order to save time and storage, highlighting in different domains is one of the hardest challenges in computer science. Things that are intuitive to human beings like ”the main scene” are inherently case-dependent and difficult for machines to internalize or generalize [3]. There are nonetheless some projects in the industry.

[2] focuses on basketball videos and it uses deep learning and Amazon Mechanical Turk. It has 4153 clips, each 10 seconds long and these clips are pointed from 0 to 3 by Amazon Mechanical Turk employees as following the instructions. It has 4 classifiers, one of them for audio and three of for videos. New clips are assigned the points as classified with data set.

Google Cloud Video Intelligence has been also analyzed. It has several features such as finding shots when the scene is changed, shot labels(20,000 labels), explicit content etc.

Microsoft Face API is chosen as the main technology used in this project.

A. Microsoft Face API

Microsoft Face API or deprecated name Emotion API has many functionalities but emotion recognition is used in this project. It presents the probabilities of the 8 different emotion in the face. These are;

- Anger
- Disgust
- Contempt
- Happiness
- Fear
- Sadness
- Suprise
- Neutrality

It has two pricing tier. The free one is used and it limits the API calls as 20 per minute.

III. Main Approach

During a conference talk, when a talker says something important or different, his/her or audience’s emotion would be differentiated from neutral. This is the main approach of the project. Therefore, the first aim is the catching lowest neutral shot.

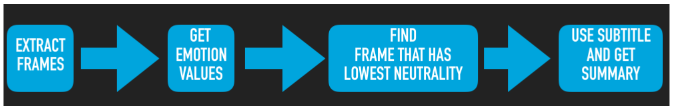
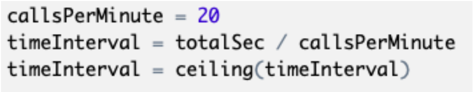

EXTRACT FRAMES
GET EMOTION VALUES
FIND FRAME THAT HAS LOWEST NEUTRALITY
USE SUBTITLE AND GET SUMMARY

Fig. 1. Flow chart

A. Extract Frames

As mentioned before, Microsoft Emotion API allows a maximum of 20 calls per minute in the free account. Thus, a maximum of 20 frames should be extracted from the video to send API.



```
callsPerMinute = 20
timeInterval = totalSec / callsPerMinute
timeInterval = ceiling(timeInterval)
```

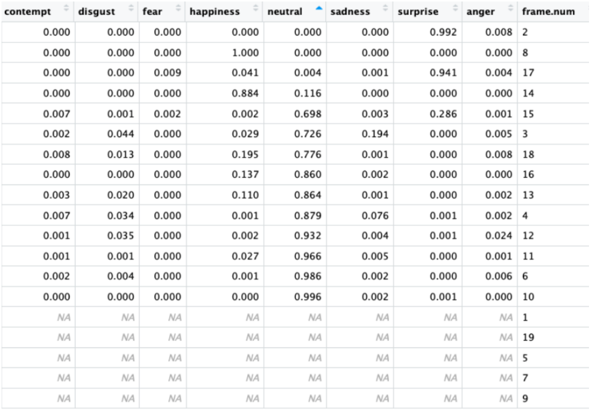
Fig. 2. Calculation of the time interval

Figure 2 shows how time interval is calculated. This value is sent to a python script to extract frames by an interval.

The Python script was very slow especially in the long videos because both reading and decoding the frame was done in the main thread [4]. In order to increase the speed, created a new thread and queue structure is built.

B. Get Emotion Values

After frames are extracted, these are sent to Microsoft Emotion API in order to get emotion values. JSON results are converted to R data frame and a table like in figure 3 is created.



contempt	disgust	fear	happiness	neutral	sadness	surprise	anger	frame.num
0.000	0.000	0.000	0.000	0.000	0.000	0.992	0.008	2
0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	8
0.000	0.000	0.009	0.041	0.004	0.001	0.941	0.004	17
0.000	0.000	0.000	0.884	0.116	0.000	0.000	0.000	14
0.007	0.001	0.002	0.002	0.698	0.003	0.286	0.001	15
0.002	0.044	0.000	0.029	0.726	0.194	0.000	0.005	3
0.008	0.013	0.000	0.195	0.776	0.001	0.000	0.008	18
0.000	0.000	0.000	0.137	0.860	0.002	0.000	0.000	16
0.003	0.020	0.000	0.110	0.864	0.001	0.000	0.002	13
0.007	0.034	0.000	0.001	0.879	0.076	0.001	0.002	4
0.001	0.035	0.000	0.002	0.932	0.004	0.001	0.024	12
0.001	0.001	0.000	0.027	0.966	0.005	0.000	0.001	11
0.002	0.004	0.000	0.001	0.986	0.002	0.000	0.006	6
0.000	0.000	0.000	0.000	0.996	0.002	0.001	0.000	10
NA	NA	NA	NA	NA	NA	NA	NA	1
NA	NA	NA	NA	NA	NA	NA	NA	19
NA	NA	NA	NA	NA	NA	NA	NA	5
NA	NA	NA	NA	NA	NA	NA	NA	7
NA	NA	NA	NA	NA	NA	NA	NA	9

Fig. 3. Emotion probabilities with example frames

C. Find Frame that has Lowest Neutrality

After creating the table of emotion values, the frame that has the lowest neutrality is found easily and it is the main frame what is used during the summarization process. However, if this frame is used as an end of the summary without any control, it may present a centre of the sentence, not a full sentence. In order to extract a meaningful section from a video, at least a full sentence should be extracted.

D. Use Subtitle and Get Summary

A summarized video has to start with the beginning of a sentence and finish with an ending of a sentence.

The duration of the summarized video is initialized as 10 seconds and it can be changed according to speech. This is going to be explained more with an example.

IV. Demo

The project needs a video and subtitle as inputs. The video and subtitle of a talk [5] were downloaded as using a 3rd party application.

Figure 3 belongs to this video.

Frame 2 is a presentation page during a talk but the system does not recognize it whether the frame from direclty speech or a slide. It assumes that the talker said something different or important, and then, the emotion of audience or talker is changed to non-neutral.

In this example, the variables are initialized like that;

- lowestNeutralFrame = 2
- approxStartingTime = 00:00:50
- approxFinishingTime = 00:01:00

As seen in figure 8, approxStartingTime is not aligned with the starting of a sentence and approxFinigshingTime is not aligned with the finishing of a sentence.

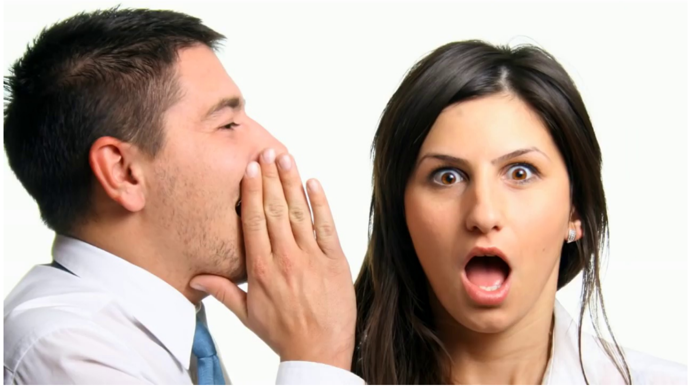


Fig. 4. Frame 2 has the lowest neutral probability with 0%

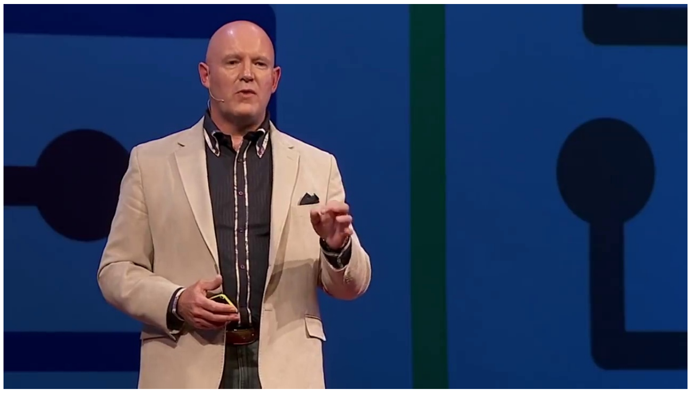


Fig. 5. Frame 10 have the highest neutral probability with 99.6%

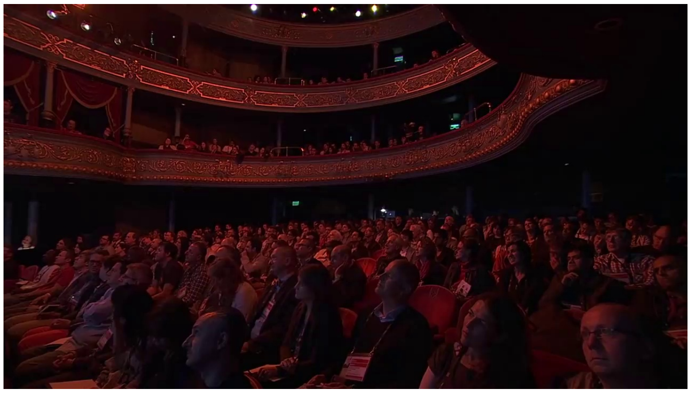
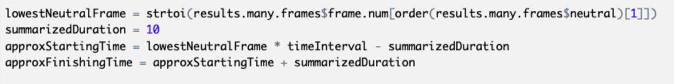
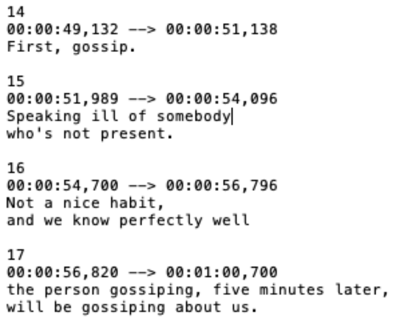


Fig. 6. Frame 1 no face recognized



```
lowestNeutralFrame = strtoi(results.many.frames$frame.num[order(results.many.frames$neutral)[1]])
summarizedDuration = 10
approxStartingTime = lowestNeutralFrame * timeInterval - summarizedDuration
approxFinishingTime = approxStartingTime + summarizedDuration
```

Fig. 7. Starting and finishing time before reading the subtitle



```
14
00:00:49,132 --> 00:00:51,138
First, gossip.

15
00:00:51,989 --> 00:00:54,096
Speaking ill of somebody
who's not present.

16
00:00:54,700 --> 00:00:56,796
Not a nice habit,
and we know perfectly well

17
00:00:56,820 --> 00:01:00,700
the person gossiping, five minutes later,
will be gossiping about us.
```

Fig. 8. The related subtitle section

Starting times of the subtitle are compared with ”approxStartingTime” and the nearest point to it is chose as summary starting time. In this example, it is going to be 00:00:49,132.

After ”approxFinishingTime”, time out of the first sentence with ending dot(.) represents finishing time of the summary. If the end of the line has a dot(.), it means that this sentence is ended and can be used to detect the last point of the summary.

Finally, the video is cut by FFMPEG. The starting time and the finishing time of the summary is like that;

- summaryStartingTime = 00:00:49,132
- summaryFinishingTime = 00:01:00,700

V. Evaluation and Future Perspective

Maximum 20 frames are analyzed per a video because of the free account limitation so if the standard account is purchased, more frames(maybe all frames, it depends on the speed) would be analyzed and the possibility of the finding lowest neutral frame would be increased.

Summarizing mechanism works differently in the human brain and it is not possible to give the key idea of a talk with around 10 seconds video. This system extracts a section based on emotions. However, the main motivation of realizing the project is achieved and it can be used to have a trial video combining short sections from different talks

This study can be improved by adding a semantic analysis. Therefore, it may present a more meaningful summary.

References

[1] YouTube. 2019. TEDx 2017 trailer - YouTube. [ONLINE] Available at: https://www.youtube.com/watch?v=BVC_eT_liqI&feature=youtu.be&t=28 [Accessed 13 February 2019].

[2] William Spearman. 2015. Using Deep Learning to Find Basketball Highlights. [ONLINE] Available at: https://medium.com/in-the-hudl/using-deep-learning-to-find-basketball-highlights-edd5e7fa1278. [Accessed 13 February 2019].

[3] 1)Bing. 2018. Intelligent Search: Video summarization using machine learning. [ONLINE] Available at: https://blogs.bing.com/search-quality-insights/2018-08/Intelligent-Search-Video-summarization-using-machine-learning. [Accessed 7 February 2019].

[4] PyImageSearch. 2019. Faster video file FPS with cv2.VideoCapture and OpenCV - PyImageSearch. [ONLINE] Available at: https://www.pyimagesearch.com/2017/02/06/faster-video-file-fps-with-cv2-videocapture-and-opencv/. [Accessed 08 February 2019].

[5] TED. 2019. How to speak so that people want to listen — Julian Treasure. [ONLINE] Available at: https://www.youtube.com/watch?v=eIho2S0ZahI&t=26s. [Accessed 14 February 2019].

[6] Github. 2019. ValuableMoment. [ONLINE] Available at: https://github.com/zazazingo/ValuableMoment. [Accessed 14 February 2019].