Adaptive subtitle allocation with speakers separation

Hojeong Lee **Hyukhee Han** **Hyunseok Choi** **Juhyeon Nam** **Kawon Lee**
Carnegie Mellon University
Pittsburgh, PA 15213
{hojeongl, hyukheeh, hyunseoc, juhyeonn, kawonl}@andrew.cmu.edu

Abstract

Captions help people understand the contents of the video. However, when there are multiple speakers at the same time, it is difficult to understand who is speaking through the subtitle. Also, most subtitles are located below the scene. Therefore, we allocated subtitles under the speaker's face. A speaker's face was cropped with face detection using YOLOv3. Then, a lip-reading model was applied to the cropped video to match ground-truth subtitles. This project allows viewers to properly recognize spoken subtitles in situations where multiple people are speaking. Implementation code can be found here: `https://github.com/AirHorizons/CUTUBE`.

Figure 1: Before and After Applying Adaptive Subtitle Allocation

1 Introduction

The subtitles in videos help viewers understand the scene by displaying a speaker's speech or even translating those into different languages. It can also make the video more dynamic and interesting in some scenes. However, current subtitles are located statically at the below part or the upper part of the frame for the entire video. The current static subtitle position has several disadvantages. First, it cannot reflect the location of the speaker. This problem can be exacerbated in scenes where multiple people are speaking simultaneously. Second, it forces viewers to look at a fixed location of the frame and distracts their attention from the scene. Furthermore, subtitles displayed in fixed positions may obscure important information in the video.

Our project aims to address these issues and introduces a dynamic subtitle allocation pipeline that can be applied to any video with subtitles. We located subtitles to a prominent position near the speaker so that they occupy the more natural and intrinsic part of the scene. We assumed that the existing videos generally do not have multi-channel audio inputs nor stereo audio inputs. These assumptions make our pipeline more widely applicable for existing videos, since it does not require any kind of additional pre-processes in the filming step. The process and approaches we implemented are

explained below.

2 Related Works

We utilized several deep-learning-based models with pre-trained weights in our system. In this section, we briefly introduce these models.

2.1 Face Detection

Face detection is a task to locate a region of human faces in the input image. Since we have to locate speakers' faces to allocate subtitles near them, we use a pre-trained model which can perform face detection task. The model is called YOLOFace[1]. As mentioned above, we utilize this to detect the speaker's face in the input video. The model is based on the model called YOLO v3[4], which is a state-of-the-art, real-time object detection model. It only retrieves the face class from the detected object by YOLO v3.

2.2 Visual Speech Recognition

Visual Speech Recognition(VSR), also called Lip Reading, recognizes speech content based on the movements of lips without audio features. VSR is used for silent speech or poorly-performed CCTV cameras that only have visual information. It can be helpful for people who cannot hear. The lip reading can also be used to recognize sentences in noisy environments. When multiple people speak at the same time, lip reading allows subtitles to be assigned to the speakers. The performance of the state-of-the-art lip reading model is better than the trained human lip readers[3].

Although the speech-to-text using audio performs better than lip reading, the prediction of a sentence when multiple speakers are talking will help allocate the captions to a specific speaker. We used the Visual Speech Recognition for Multiple Languages [3] based on ESP-net, which provides the state-of-the-art algorithms for end-to-end visual speech recognition in the wild, with the pre-trained model with The Oxford-BBC Lip Reading Sentences 3 (LRS3) Dataset[2]. We can use this to extract sentence predictions.

3 Methodology

The initial pipeline of the task is identifying different sources of speech and separate them accordingly. The baseline model is proposed by Afouras et. al[1]. Since visual speech recognition only takes a video with one speaker, a face recognition model was used before that to create a video where only one person was cropped from the original video.

3.1 Cropping Single-speaker Videos

First, we need to crop a single-speaker video from the original input video to distinguish which speaker is speaking a particular line in the original subtitle. There are two steps to do so: 1) face detection in each frame and 2) face-cropped video generation.

For the first step, face detection, we utilized a face detection model with pre-trained weights called YOLOFace. YOLOFace model takes a single frame image as input and returns a set of locations of all faces in that image in a format of bounding boxes.

In the second step, the face-cropped video generation, we concatenated the cropped images of detected faces in the first step. However, the problem here is that the YOLOFace only detects multiple faces in the video and does not verify which of the detected bounding boxes belongs to the same person compared to the previous frame. We may solve this problem with a face verification model, but it will be computationally expensive to perform on all frames in the input video. So we use a relatively simple distance-based approach to determine which detected face belongs to the same person among the previously detected faces. For a distance measure, we summed up an L1 distance

[1]The code and pre-trained weights can be found at `https://github.com/sthanhng/yoloface`

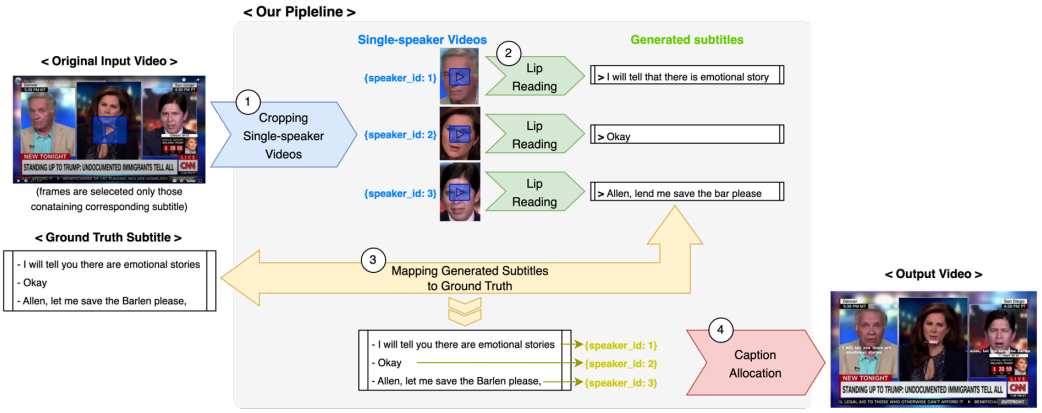


Figure 2: The overall pipeline of Adaptive Subtitle Allocation

between all corners of bounding boxes in the previous frame and the current frame as represented in the equation 1.

$$\text{Speaker ID}_{t,i} = \underset{j}{\text{argmin}} \sum_{k=1}^{4} (|x_{t,i}^{k} - x_{t-1,j}^{k}| + |y_{t,i}^{k} - y_{t-1,j}^{k}|) \tag{1}$$

The equation explains how we define a speaker id of the ith face in current frame t, $\text{Speaker ID}_{t,i}$, where $x_{t,i}^{k}$ and $y_{t,i}^{k}$ denote the x and y coordinate of kth corner of the ith bounding box in frame t, respectively. In summary, we calculate the distances between all pairs of faces in subsequent frames and consider those with the closest distance as the same person. Additionally, we have done another post-processing to the bounding boxes of detected faces. Since the sizes of detected bounding boxes are vary through the frames, it cannot be concatenated right away. Therefore, we calculate the maximum size of bounding boxes in the whole frame for each speaker and slide the maximum-sized bounding box based on the center point of the originally detected bounding box in every frames.

3.2 Lip Reading

Now, we have videos of each person cropped during the time shown in each caption. Each of these video clips was lip-readed using a pretrained model based on Lip Reading Sentences 3 (LRS3) Dataset which is consists of thousands of spoken sentences from TED. LRS3 dataset is the largest publicly open audio visual english data set that has over 400 hours of video data. After Lip Reading, we got CSV file that has predicted result of each subtitle time.

3.3 Mapping the Generated Subtitles to Ground-truth

After lip reading, the model has *N* reconstructed sentences, and *N* ground truth subtitles. The task is providing a correct mapping based on the similarity of the sentences from the two domains. The mapping will be chosen to maximize the aggregated similarity(or minimize the total distance). We have investigated two methods with two input forms of sentences.

3.3.1 Edit Distance versus Dynamic Time Warping

The two algorithms regard a similarity of two sequences as a minimal cost of changing one sentence to another. The cost of adding, deleting, and modifying data in conventional edit distance(Levenstein Distance) is equal to 1. Dynamic Time Warping differs from the edit distance in that the cost is zero for adding or deleting a duplicate of a time step or adding or deleting zero for both ends. This makes the cost invariant for both globally and locally time-shifted series. However, the DTW algorithm assumes that elements of the sequence are in a metric space, which means the distance between two elements should be derived somehow. Currently, we implemented a discrete distance function that returns 0 if the two symbols are equal and 1 otherwise.

3.3.2 Lexemes and Phonemes

Typically, the unit of the sentence under the measurement is the lexeme. We use the raw characters to calculate the distance between the ground truth subtitle and the generated sentence. This would be fine when the reconstruction is always perfect. Nevertheless, in reality, lip reading can end up making wrong words with similar pronunciation, such as 'syllable' and 'pinnacle'. The problem is emphasized more when the vocabulary for the reconstruction domain is relatively small compared to the ground truth. So we decided to phonemize sentences before retrieving distances.

We defined the score as 1 - (tr A / sum A) for the confusion matrix A. The score is higher if the diagonal element (the distance of correctly labeled sentences) is smaller than the others. As expected, phoneme distances yielded better results compared with lexeme distances. However, DTW underperformed the conventional edit distance for the unit test dataset. This is due to an incomplete algorithm translation into the sentence domain. Since the element-wise cost is a constant real value in the original algorithm, we must distinguish cost value favoring phonemes with similar pronunciation (such as θ and t). Also, data from our experiments on comparing algorithms were short video clips with single sentences and single speakers, randomly choosing N instances since the sentences are independent of each other.

Thus, the DTW algorithm has performed better when applied to real multi-speaker video because it shows more robustness in several aspects. First, real-life multi-speaker videos have faces that do not speak. Our model has additional code to cope with the mismatching number of subtitles and lip reading generations, but this leads to performance degradation if blindly applying edit distance. Second, the sentences played simultaneously mostly contain overlapping words since they speak for the same topic. The above concerns require the algorithm to act more robustly, giving DTW more performance on the actual data.

3.4 Caption Allocation and Combining

There were two additional problems in caption allocation in our project. One was the fluctuation of the allocated caption and the other was dealing with the subtitle length. At first, we assigned each matched subtitle to right lower part of speaker's bounding box point. However, as the bounding box varies over time, the position of the subtitle also fluctuates. To solve this problem, the position of the subtitles on the output screen is fixed to the relative position of each speaker. The other problem was the subtitles were too long if we allocated them in one line. So, we make the subtitles into multiple lines with six or seven words in each line. Finally, the processed video as visual data and the audio of that video are merged into the final output.

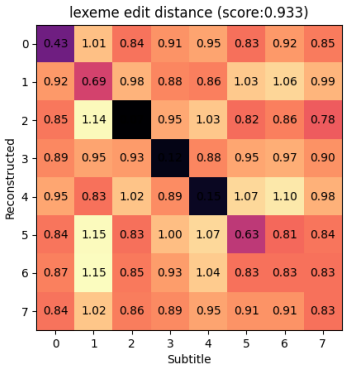
4 Results

4.1 Comparison

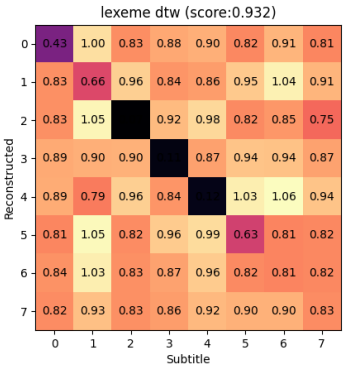
In the Figure 4 and 5, the pictures in top row are captured from original videos and those in bottom row are captured from our result videos. Even though there are some mis-matched subtitles, almost subtitles are assigned and positioned appropriately. When several people speak at the same time, subtitles occupy a large part of the screen and can occlude the speaker. Also, when the sound is not clear, such as when there is noise around viewers, it is difficult for the viewers to understand what each speaker is saying in the video. However, as you can see the resulting video, the subtitles are separated and assigned to each speaker so that the viewers can understand them much more easily.

4.2 Accuracy by Comparing with Ground-Truth

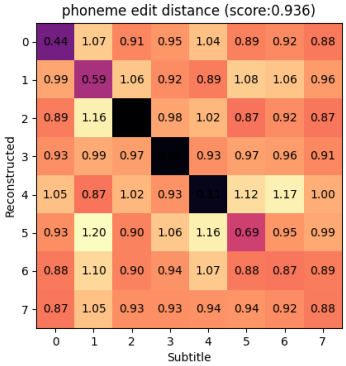
We have matched the assigned lines with ground truth and calculated their accuracy. The video with 2 speakers has 153 seconds of duration and 35 lines in the whole, and 22 out of them are assigned to the correct speaker. So, the accuracy of our result is around 62.86% for this video. On the other hand, the another video where 3 people speak simultaneously has only 11 seconds of duration and 8 lines, and all of the lines are matched to the right speaker. Hence, the final accuracy is measured as 69.77%, which is calculated by 30/43.



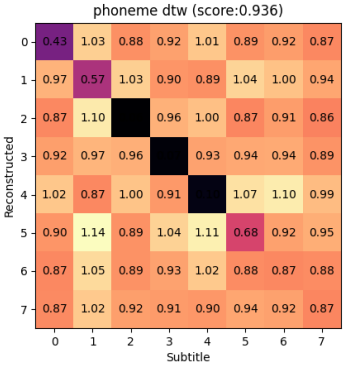
(a) lexeme edit distance



(b) lexeme DTW



(c) phoneme edit distance



(d) phoneme DTW

Figure 3: Distance between subtitles from video data with measuring (a) lexeme edit distance, (b) lexeme DTW, (c) phoneme edit distance, and (d) phoneme edit distance. The axes mean the label of video files and values stand for distance of each pairs.

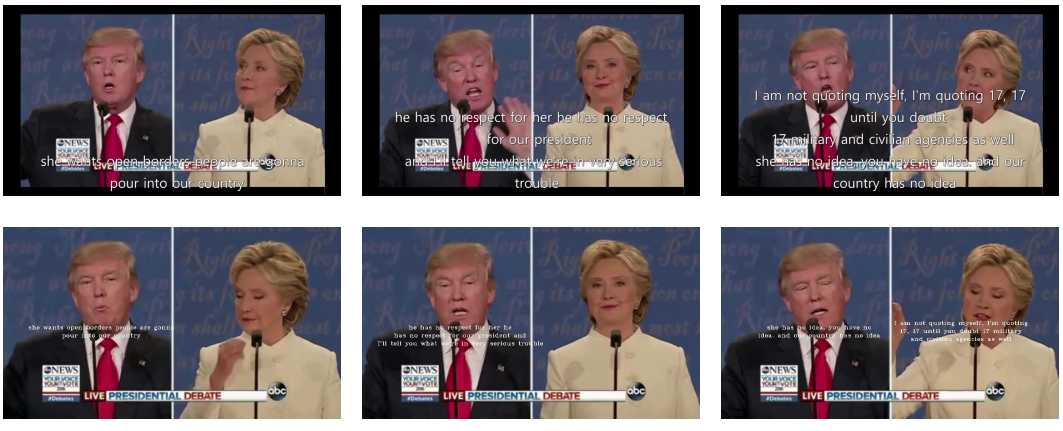


Figure 4: video with 2 speakers

Figure 5: video with 3 speakers

5 Furthermore Research

5.1 Automatic Speech Recognition

Our final model requires ground truth subtitles so that it can match the text with generated subtitle via lip reading. The necessity of this additional data originates from inaccuracy of lip reading, where the word level average recall is less than 50% of the whole sequences and when it comes to sentence level, there was literally none. For reconstructing text with high precision, Automatic Speech Recognition(ASR) is a viable option. Conventional ASR system aims to convert single speaker speech into the text, but recent research[5] yields reasonable result from multispeaker environment.

5.2 Dynamic Text Style Adaptation

By adjusting the color and size of the subtitles more expansively, the subtitles can be lively. When speaker detection and audio separation are completed, we have an audio clip consisting of only one voice. Therefore, colors can be assigned depending on a specific person, and the size of the subtitle is adjusted according to the volume.

5.3 Allocating Generated Subtitles in the Video with Optimal Placement

In our previous report, we planned to place subtitles in the appropriate locations where they do not obscure essential parts of the video. However, our final model works without the smart locating algorithm. Although we have placed each subtitle under the center of each speaker, there is still room for improvement in determining the optimal positioning of the subtitles. For example, using existing computer vision techniques to find the best position using the box's transparency and background color.

References

[1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. The conversation: Deep audio-visual speech enhancement, 2018.

[2] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Lrs3-ted: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496*, 2018.

[3] Brais Martinez, Pingchuan Ma, Stavros Petridis, and Maja Pantic. Lipreading using temporal convolutional networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6319–6323. IEEE, 2020.

[4] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018. cite arxiv:1804.02767Comment: Tech Report.

[5] Robin Scheibler, Wangyou Zhang, Xuankai Chang, Shinji Watanabe, and Yanmin Qian. End-to-end multi-speaker asr with independent vector analysis, 04 2022.