News Video Story Segmentation based on phonetic and visual features

Lin Jialiang*, Feng Qingduo, Shen Zhentao, Lin Anqi

Computer Science and Engineering Department
Hong Kong University of Science and Technology, Hong Kong
** jlinbi@connect.ust.hk*

***Abstract* — In recent years, automatic news categorization has become an important research topic due to the needs of saving time and the huge amount of Internet videos. This paper proposes a news splitting algorithm based on machine learning algorithm, phonetic characteristics and visual features. Extracting a total of seven-dimensional features from the video sampling nodes, and using the random forest algorithm as a classification model, to train and test the news split model. The seven-dimensional features are as follows: (textual:) semantic similarity scores, semantic depth scores, presence or absence of keywords, the time interval between two nodes, (image:) whether it is a live studio footage, with or without of live and non-live studio-inter-lens switching, with or without lens switching. The implement of feature extraction from speech text refers to the algorithm related to the field of text segmentation technology, such as TextTiling model, word2vec and so on, and the image feature extraction refers to the algorithm related to the field of shot boundary detection. In the experiment evaluation, a set of 100 CCTV news videos containing tens of thousands of candidates split points are used as data sources. Experiment shows that the entire model achieve average precision 0.93, average recall 0.87.**

***Index Terms* -- Automatic segmentation of news items; Text segmentation; Shot boundary detection; Random forest.**

I. INTRODUCTION

A. Project Background

With the rapid growth of the access to the Internet, there is an increasing number of videos online. And digital news video is one of the convenient media for news acquisition [1]. However, video is more time consuming than reading since it requires more watching and listening. So, video segmentation and summarization through video processing technology raise people's great interests.

The trend of information fragmentation has gradually penetrated into the video field. In various industries, the demand for news video storage and information mining has gradually increased. How to split a long news videos into multiple news items of different topics has become a very hot study.

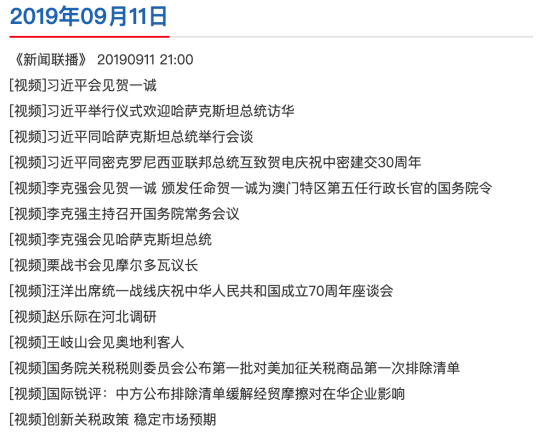


Figure 1. Entire news video has been segmented into several

Video not only provides rich visual information from the image, but also provides rich phonetic information. The speech content in news also plays a vital role in news segmentation. Therefore, we can capture these features to train a classifier to detect whether it is a split point at the timestamp.

B. Project Description

The news video fragment can be divided into the following three categories:

Figure 2. Three categories of nes video fragment

From the figure, it is clear that we need to find the spilt points which can divide the news fragment. Therefore, what we are going to do is to: (1) Find the candidate split points and then (2) classify them into 0 or 1, which turns into a classification problem.

As for goal (1), the candidate split points is defined as the timestamps which separate the image content or the timestamps that hosts begin to talk a new sentence. Comparing with these two definitions from two different sources, we can know that the latter one is clearer and more straightforward.

As for goal (2), we need to extract some features from the audio and image to form a feature vector representing the candidate split points [2]. Then we need to train a classifier to determine the mapping of these feature vectors, thus distinguishing these candidate points are true split points or not.

According to the above analysis, we design the whole structure as the following figure:

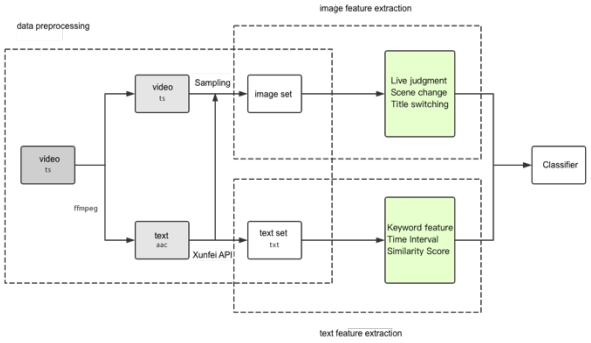


Figure 3. Whole structure of news video story sengmentation

The whole structure can be described as four parts: video data preprocessing, image feature extraction, textual feature extraction and classifier.

This article documents our whole model construction, experimenting process, and discuss the evaluation and applications. In section 1, we introduce the project background and core idea of the project as well as the entire structure. In section 2, the phonetic feature extraction is illustrated. In section 3, the visual feature extraction is illustrated. In section 4, we introduce our classifier and the experimental effects. In section 5, we express our thanks to prof. and TA as well as giving individual contribution of the project.

II. PHONETIC FEATURE EXTRACTION

A. Data Preprocessing

From the Figure 3, we can firstly obtain acc audio file from ts video file with the tool ffmpeg [3]. Then we can use iFLYTEK's API [4] to transform audio to text, getting the json result. In the json result, each meta corresponds to one sentence, with the beginning timestamp, which we can utilize to sample images from video. After these data preprocessing procedures, what we have are the textual information of each candidate points and sample images.

Since the codes of utilizing the tool ffmpeg and the iFLYTEK's API are attached in the code zip file and they are not complex actually, not related to our core design as well, we briefly illustrate them in this way.

B. Textual feature extraction

1. Chinese word segmentation

Unlike English words, Chinese words do not have natural distinguishing way to identify them. Therefore, we need to utilize python package jieba [5] to solve this problem.

Chinese word segmentation refers to the segmentation of a Chinese character sequence (sentence) into individual words, and the final sentence can be expressed as a collection of multiple words. Only after doing this, can we train the corresponding Chinese word vectors.

2. Chinese word vector

Chinese word vector construction, that is, according to the unified conversion rules, each Chinese word is uniquely assigned a vector representation, so that the machine can read it. In the field of natural language processing, word vectors are used to identify and preprocess words, and then the corresponding algorithm is calculated. Word vector needs to meet the similar requirements of words with similar semantics in word vector space, which is particularly important for the application of text classification, text clustering and other fields. This project uses the Chinese word vector table constructed by word2vec [6].

3. Word filtering

Text feature extraction is mainly based on the text content. Before calculating the similarity between sentences, we first filter the words in the sentence "one best". Based on the improvement of TextTiling [7] algorithm and referring to the concept of document dependent stop words proposed by Xiang Ji Hongyuan Zha [8] and others in another paper, this paper determines the types of words to be filtered out: high frequency words, punctuation marks, dependent forbidden words. High frequency words, such as "的", "了", "会" and other words that repeat more than 1/3 of the total number of words. Dependency disallowed words: these words are very useful in distinguishing several different documents. Generally, they can summarize the subject words of the whole document, but they are disadvantageous for detecting sub topics in the document. In this project, high frequency words evenly distributed in the document are regarded as dependent forbidden words. The purpose of lexical filtering is to transform sentence into a set of useful participles

4. Textual features

4.1. Similarity score

According to TextTiling [7][9] algorithm, the semantic similarity between sentences is calculated. First, the semantic similarity between words is calculated by cosine similarity formula:

$$sim\_words(x,y) = \frac{\sum_{i=1}^{n}(x_i * y_i)}{\sqrt{\sum_{i=1}^{n}(x_i)^2} * \sqrt{\sum_{i=1}^{n}(y_i)^2}}$$

As for block partition and window construction, according to the TextTiling algorithm, it is necessary to block the news text composed of "onebest" set. Experiments show that the system performance is the best when the block size is 6, that is, 6 statements form a block. Arrange "onebest" in order to form news documents and form an interval point between each two consecutive "onebest". Take the interval point as the symmetry center and the size as the block size to construct windows and calculate the semantic similarity score at the interval point.

In this algorithm, the block size is not fixed. The reason is that there is only one sentence on the left side of the first interval point. If it is compared with six sentences on the right side of the interval point, it is obvious that the information is asymmetric. Therefore, when the number of sentences on the left and right sides of the interval point is inconsistent, it generally refers to the beginning and the end of the document. Then the minimum number of sentences is used for unified comparison on both sides. This eliminates the influence of the deviation of similarity calculation in the beginning part of the document on the results of the algorithm:

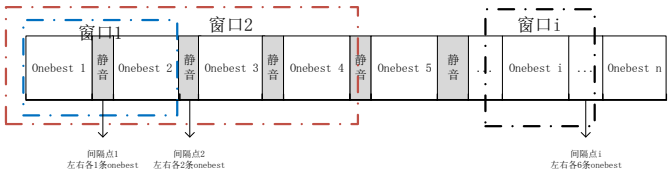


Figure 4. Block partition

Therefore, the calculation formula of similarity score is:

$$\mathrm{sim_b locks}(b_1, b_2) = sum/(n+m)$$

Where $sum = \sum_{i=1}^{n} \max\left(sim\_words\left(T_{b_1,i}, T_{b_2,j=1\ldots m}\right)\right) + \sum_{j=1}^{m} \max\left(sim\_words\left(T_{b_1,i=1\ldots n}, T_{b_2,j}\right)\right)$

4.2. Deep score

Draw a graph of similarity score. The abscissa is the interval time, and the ordinate is the semantic similarity score of the current interval. The local highest point of the curve represents the maximum similarity of text content on both sides of the interval point, and the local lowest point represents the minimum similarity of text content on both sides. For each interval point, for example, gap_{18} finds a local highest point gap_{15} to the left and a local highest point gap_{19} to the right. The depth value is calculated as follows:

$$\mathrm{Depth}_{18} = gapscore_{15} - gapscore_{18} + gapscore_{19} - gapscore_{18}$$

The depth score can reflect the relationship between the two adjacent text content. The greater the depth value is, the lower the similarity between the two sides of the text content is; on the contrary, the smaller the depth value is, the greater the similarity is. Fig. 5 and Fig. 6 are the charts calculated from the node similarity score and depth score of the same CCTV news video.

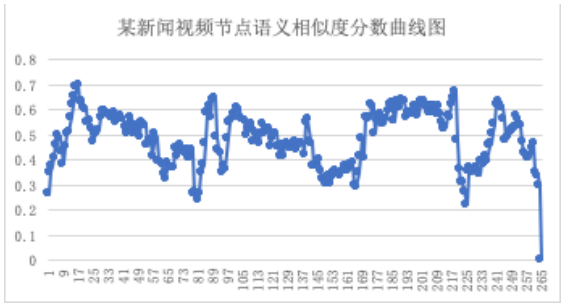


Figure 5. similarity score

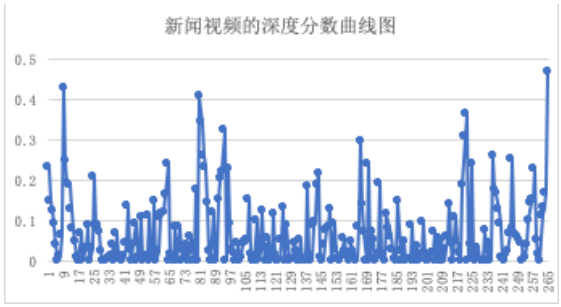


Figure 6. deep score

4.3. Keyword

For news broadcast in CCTV1, the key words are as follows: "国际快讯"、"联播快讯"、"详细报道"、"详细内容". Judge whether "onebest" sentence contains keywords and mark them.

4.4. Time Interval

By calculating the difference between beginning timestamp of this sentence and the ending timestamp of last timestamp, we can distinguish whether a “post” in it or not.

III. VISUAL FEATURE EXTRACTION

A. Independent Title Classification

1) Single Gaussian Background Modeling

Single Gaussian Background Modeling is also called as Background estimation. It is operated on sequential pictures (continuous frames, like video), transforming a object detection problem into a binary classification task. It compares the pixels in current frame with the mean values of each pixel calculated from previous frames. And then all pixels are separated into 2 classes, background and foreground, through a Gaussian Distribution updating with the previous pixel-values.

2) Motivation

The subtitle always stably appears at the bottom of the frame and are always with background in pure color. Actually, Single Gaussian Background Modeling are sensitive to light variation and complex background. But, in our task, these kinds of challenges are not existed. Therefore, Single Gaussian Background Modeling can be used to classify whether the

subtitle is a title for a segmentation of the video (i.e. is not a normal subtitle to display what the people in the video is talking about). The background at the bottom of each frame detected by the Single Gaussian Background Modeling will be a title which is the summary about current segmentation of video.

3) Algorithms

Single Gaussian Background Modeling is a method to extract and update Background and Foreground during the detection of moving object.

It supposes that the brightness of pixels for background satisfies Gaussian Distribution. i.e. for background B, the brightness of each pixel (x, y) satisfies $B(x, y) \sim N(\mu, d)$:

$$p(x) = \frac{1}{\sqrt{2\pi}d} \, exp^{-\frac{(x-\mu)^2}{2d^2}}.$$

All the pixel point has 2 attributes: mean value μ and variance d. The mean and variance values are calculated through the accumulation on all the frames in that specific period. These 2 attributes form the representation of the background model.

For each pixel, if

$$\frac{1}{\sqrt{2\pi}d} \, exp^{-\frac{(G(x,y)-B(x,y))^2}{2d^2}} > T$$

, it is a Background point, or it is a foreground point. T is a constant threshold and G is a current frame.

After the classification part, each pixel will be updated by

$$B_t(x, y) = p * B_{t-1}(x, y) + (1 - p) * G_t(x, y)$$

, in which p is a constant, as the updating rate (bigger p, slower update).

B. Studio Background Generative Model and Classification

1) Motivation

This part makes use of the low-level features: color and color's statistic features. Generate the model of studio, and compare each frame with it to classify current frame is studio or not.

2) Algorithms

a) Blur

$$1/8 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

We choose the 8-neighbor mean template for convolution to flat the edge, reducing the details to stress the major color in a specific region.

b) Statistic

We count the number of each color and find out the color with largest number. And calculate the percentage of the color closed (±30) to it. Through observation, the major color should be blue, and it's percentage should be around 66%.

3) Application

The 386th frame is classified as not a studio. The 670th frame is classified as studio.

Figure 7. Studio Background Generative Model Application 1

These two frames showed in Figure 7 and 8 fit the generative background model of studio, but it is obviously incorrect. We will correct it through the latter part, detecting the hosts' faces to correct these kind of frames are not in studio-background. Because the hosts are always in the studio.

Figure 8. Studio Background Generative Model Application 2

C. States of Hosts Recognition

1) Motivation

Detection of human face is much easier than the recognition of a specific person's face. There are two reasons, 1) when the number of training samples is small, there will be not enough negative samples for the classifier to train to distinguish the

correct face from other faces. 2) a pre-trained model cannot scale itself to a new sample, requiring re-training for the model.

These are kind of one-shot problem. And for our task, there are only 2 hosts and different hosts for each video. So, it is in the same situation to the one-shot problem.

Therefore, we have Siamese Network for this task. This network is trained to measure similarity between 2 input pictures.

2) Algorithms

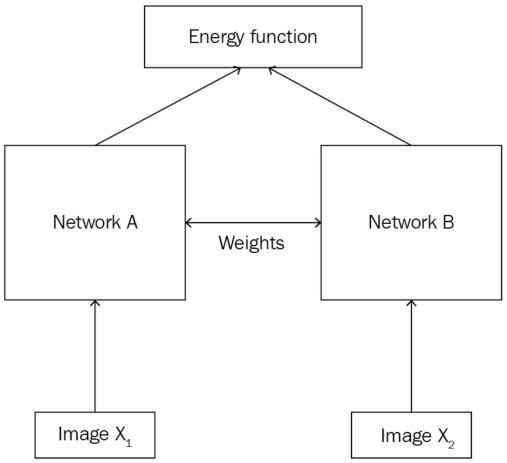


Figure 9. Constucture of Simese Netword

In Siamese Network, there are two same networks with shared weights.

In our task, these two networks are CNN network which is good for Computer Vision application, digging the useful features for current application.

The output of Network A is:

$$F_A(X_1) = CNN(X_1)$$

, and the output of Network B is:

$$F_B(X_2) = CNN(X_2).$$

The energy function is:

$$Energy = ||F_A(X_1) - F_B(X_2)||.$$

The whole model is learnt with a principle: when the input images X_1 and X_2 are the same, the output of energy function should be smaller than a constant threshold; when these two images are different, the output should be larger than the constant threshold.

D. Subtitle Extraction

OCR, Optical Character Recognition, is to analyze, recognize and obtain the text in pictures, returning the text form of the characters in the input picture.

The general structure is like Figure 10 :

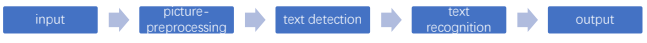


Figure 10. Structure of OCR

1) Picture-preprocessing

CNN-based deep learning technique (LeNet-5) is used to extract the features in the input picture. It is robust to the blur, disform, cluster background and brightness variance. Augment the original picture and have positive improvement to the following operation.

2) Text detection

The aim is to detect the position, range and layout of the text. Faster R-CNN uses RPN (Region Proposal Networks) as help. The algorithm is consisted of 2 parts, shown in Figure 11: 1) RPN classifies the candidate bounding-box is target or not; 2) classify the class of the target. The text in our task are always located in a limited region with pure-color background, so this algorithm performs well.

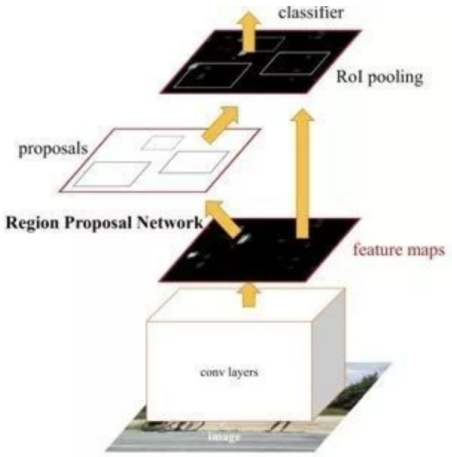


Figure 11. Constructure of Fast R-CNN

3) Text recognition

The aim is to recognize the text and translate it into correct text or specific words.

According to the CRNN model, it uses CNN features as the input, following the bi-directional LSTM to do the sequential process. The classification model produces the feature map. The output result is translated by Connectionist Temporal Classification (Neural Network-based Sequence Classification).

IV. CLASSIFIER & EXPREIMENT

A. Classifier

1. Feature Engineering

Values of Attributes from text and image processing are various, with types of binary, float, etc. We performed data analysis and further some feature engineering in order to accelerate the model convergence speed and improve the accuracy. The main processing method if oversampling.

Oversampling is technique used to adjust the class distribution of a data set. Imbalance of binary data would affect the output of a classifier because some models classify by

threshold, so the default threshold would cause the model to output biased result.

In the dataset of our project, the ratio of Class 1 and Class 0 is 0.05, which is severe imbalance, so we duplicate data with Class 1 to make ratio close to 1, rather than randomly oversampling, which will cause overfitting easily

2. Random Forest

Random Forest is an ensemble supervised learning algorithm for classification, it consists of a large number of individual decision trees and each individual tree in the random forest spits out a class prediction, random forest gets its final result by each tree's voting. It always has high accuracy and fits the dataset without much preprocessing well due to the use of ensemble method.

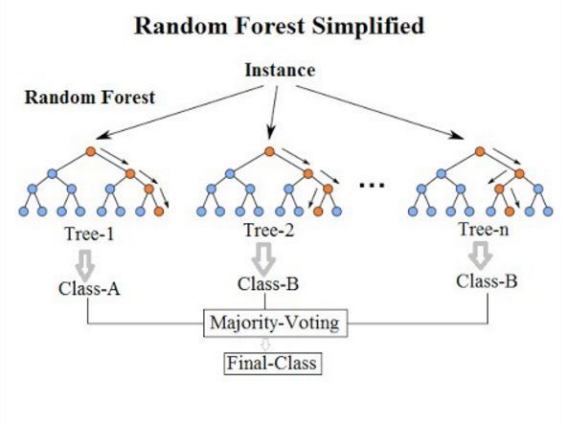


Figure 12. Example of Random Forest

As for decision tree, it is a simple classifier, it inducted rules from a set of data with label and displayed using tree structure. The advantage of this algorithm is that the rule is explainable and understandable, especially for those binary attributes.

The procedure of Random Forest is as follows: Use the Bootstraping method to randomly take m samples for n_tree times and generate n_tree training sets. Train n_tree decision tree models separately using above training sets, so each tree will have different branches due to the difference of information gain. Finally, the generated multiple decision trees form a random forest

B. Experiments

1. Dataset

In our experiments, we downloaded the CCTV news video files from CCTV official websites [10], and then get image datasets and text datasets by parsing from the video files. The Dataset preprocessing procedure is shown in Figure 3.

As for text datasets, broadcast manuscript actually, we used ffmpeg program [3] to extract it from the video, and then used Xunfei Speech Recognition API [4] converting audio data to text data in json data structure. The extracted text data contains the sample time, speaker series number and speech content, the data structure is shown in Figure 13.


```
{
    "bg":"0",
    "ed":"640",
    "onebest":"好。嗯",
    "speaker":"0"
},
```


Figure 13. Data Structure of Converted Text Data

As for image datasets, video screenshot actually, we just sample the image from the video files on the time corresponding to the time of converted text data above, in order to maintain uniformity during classifier processing.

2. Coding details

The arrangement of code is shown in Table 1.

Name	Description
Data	The Directory of data set including input data and converted data
Tool	The Directory of tools used in the project, like encoding tool
Classifier.py	Definition of classifier model definition, training.
Exfeature.py	Extract textual features
Meta.py	Candidate point object
Test.py	Main python script to get the features as csv file output
Image	The Directory of Image Processing code

TABLE I. DIRECTORY OF THE PROJECT

As for model codes, the Random Forest Classifier is implemented using sklearn package.

3. Results

We split our dataset into training dataset and validation dataset using hold-out method, and used 70% as training set, 30% as validation set. After training in the classifier, the classifier performed well in validation set. The accuracy in validation set is 1.0, which means classify the data totally correct.

4. Application

we develop a website including project basic description to display our experiment results.

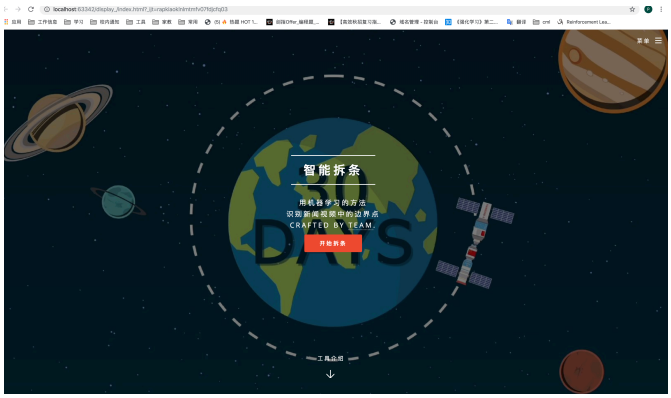


Figure 14. Project Visulization website

After inputting the date of New Feeds you want to split, the website comes up this page. The video is split according to audio and images information. The brief summary of each part is displayed by timeline and the mapping video part can watch by clicking it.

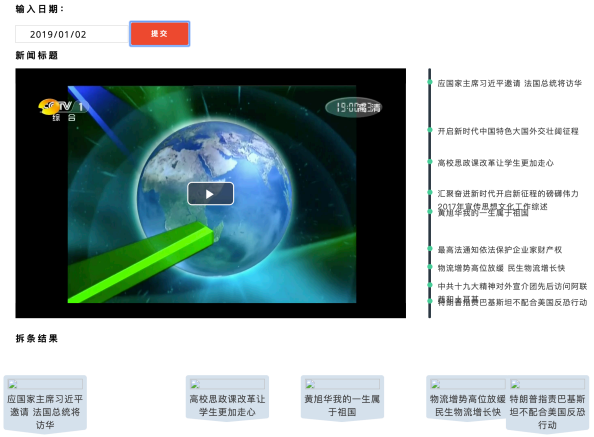
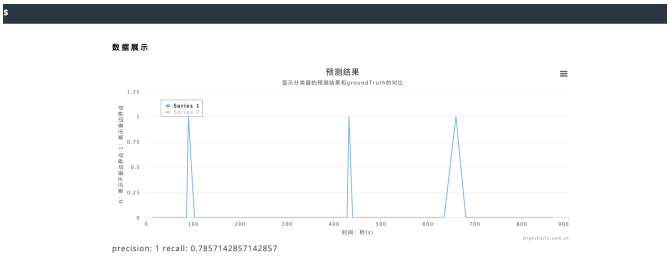


Figure 15. Visulization of News video segmentation

Here is the data display section. We draw a line chart of the predicted value and the true value at each time point, where series1 represents the predicted value and series2 represents the true value. By comparing the two curves, we find that the fitted prediction results are consistent with the true results. In addition, we show the predicted precision and recall. All this shows that this prediction is successful.



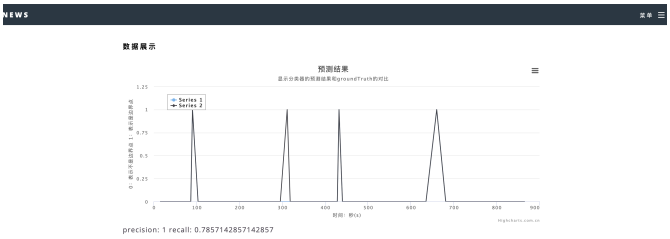


Figure 16. Examples of groundtruth and prediction (part of video)

V. ACKNOWLEDGMENT

We gratefully acknowledge the instruction and support by Prof. Nevin L. Zhang and TA TIAN Zhiliang, feeling grateful to take this course.

Here is our individual contribution:

Name	Contribution
Lin Jialiang	Data preprocessing & Textual feature extraction
Lin Anqi	Data preprocessing & Image feature extraction
Shen Zhentao	Data collection & Model evaluation and visualization
Feng Qingduo	Data collection & Classifier Construction and Model evaluation

REFERENCES

[1] J. F. Fuller, E. F. Fuchs, and K. J. Roesler, "Influence of harmonics on Zhu W, Toklu C, Liou S P. Automatic News Video Segmentation and Categorization Based on Closed-Captioned Text[C]//ICME. 2001

[2] Hauptmann A G, Witbrock M J. Story segmentation and detection of commercials in broadcast news video[C]//Proceedings IEEE International Forum on Research and Technology Advances in Digital Libraries-ADL'98-. IEEE, 1998: 168-179.

[3] ffmpeg Program. http://ffmpeg.org

[4] Xunfei API. https://www.xfyun.cn/services/lfasr?ch=bdtg

[5] Jieba Package https://github.com/fxsjy/jieba

[6] Song Y, Shi S, Li J, et al. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 2018: 175-180.

[7] Hearst M A. TextTiling: Segmenting text into multi-paragraph subtopic passages[J]. Computational linguistics, 1997, 23(1): 33-64.

[8] Gries S T. Dispersions and adjusted frequencies in corpora[J]. International journal of corpus linguistics, 2008, 13(4): 403-437.

[9] Kern R, Granitzer M. Efficient linear text segmentation based on information retrieval techniques[C]//Proceedings of the International Conference on Management of Emergent Digital EcoSystems. ACM, 2009: 25.

[10] CCTV News Dataset. http://tv.cctv.com/lm/xwlb/

[11] Breiman L. Random forests[J]. Machine learning, 2001, 45(1): 5-32.

[12] Goyal A, Punitha P, Hopfgartner F, et al. Split and merge based story segmentation in news videos[C]//European Conference on Information Retrieval. Springer, Berlin, Heidelberg, 2009: 766-770.

[13] Zlitni T, Bouaziz B, Mahdi W. Automatic topics segmentation for TV news video using prior knowledge[J]. Multimedia Tools and Applications, 2016, 75(10): 5645-5672

[14] Wang W, Gao W. Automatic segmentation of news items based on video and audio features[J]. Journal of Computer Science and Technology, 2002, 17(2): 189-195.