Deep Relationship Analysis in Video with Multimodal Feature Fusion

Fan Yu[1,2], Dandan Wang[1], Beibei Zhang[1], Tongwei Ren[1,2,*]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]Shenzhen Research Institute of Nanjing University, Shenzhen, China
yf@smail.nju.edu.cn,hedda_stone@hotmail.com,zhangbb@smail.nju.edu.cn,rentw@nju.edu.cn

ABSTRACT

In this paper, we propose a novel multimodal feature fusion method based on scene segmentation to detect the relationships between entities in a long duration video.Specifically, a long video is split into some scenes and entities in the scenes are tracked. Text, audio and visual features in a scene are extracted to predict relationships between different entities in the scene. The relationships between entities construct a knowledge graph of the video and can be used to answer some queries about the video. The experimental results show that our method performs well for deep video understanding on the HLVU dataset.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**.

KEYWORDS

Deep video understanding; relationship analysis; multimodal analysis

ACM Reference Format:
Fan Yu[1,2], Dandan Wang[1], Beibei Zhang[1], Tongwei Ren[1,2]. 2020. Deep Relationship Analysis in Video with Multimodal Feature Fusion . In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3394171.3416303

1 INTRODUCTION

A deep analysis of relationships between different entities in a long duration video contributes to deep video understanding (DVU), which requires the analysis of known information to reason about hidden information. Some tasks related to video understanding include video summarization [8], group activity recognition [10] and video visual relation detection [9]. The DVU task aims to populate a knowledge graph of a long duration video with example screenshots and types of the entities. A new HLVU dataset [2] is constructed for DVU task. The input of this task is a video with provided entity types and screenshots, and the output of DVU task is a knowledge graph. According to the knowledge graph, three different type of queries can be answered automatically. The definition of DVU task is shown in Figure 1.

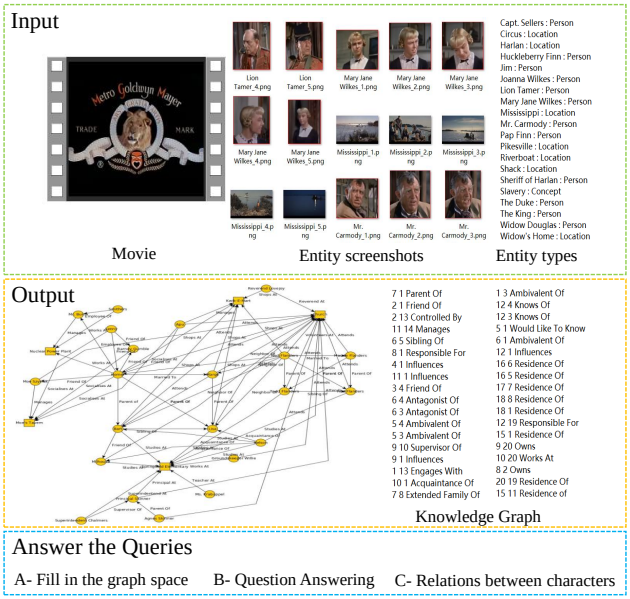

Figure 1: The definition of deep video understanding task.

DVU task needs to tackle several key challenges: (1) It is hard to extract features from long duration videos and relationships between entities may change over time. (2) Screenshots of an entity are subject to variation. For example, some screenshots of a person might be faces in close-up, while others may contain the full body. Also, the screenshots of a location may vary from the indoor scene to the outdoor scene. (3) To make deep analysis, multimodal information should be used but it is hard to extract and mix validated multimodal features. (4) The HLVU dataset provides annotations of 10 videos (6 for development and 4 for test) and such few samples adds more difficulties for training a effective model. Moreover, some relationships do not appear in the development set and how to detect these relationships in the test set is a challenge. (5) It is often difficult to distinguish between the relationships that have similar meaning. For example, it is

*Corresponding author.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3416303

difficult to distinguish the relationship "Knows In Passing" from "Knows Of". (6) Additional common sense might be used for predicting relationships between entities, especially those that have not co-occurred or even appeared in sight.

To address the above-mentioned difficulties in analyzing long duration videos, we propose a multimodal feature fusion method based on scene segmentation. The given video is first divided into separate scenes and the original frame of each generated screenshot is acquired through surf extraction and matching. After the combination of the matching result and additional person tracking, traces of every entity is then recorded and visual features of every entity can be extracted according to the union bounding box trace. We also extract audio features and text features from the audio of each scene. Our method integrates visual, audio and text features as the representation of each relationship between two entities. After multimodal feature fusion, we compute the similarity between the integrated features and the features of pre-defined relationship descriptions to obtain the relationship candidates in a scene. Finally, relationships in all scenes construct the knowledge graph of the whole video.

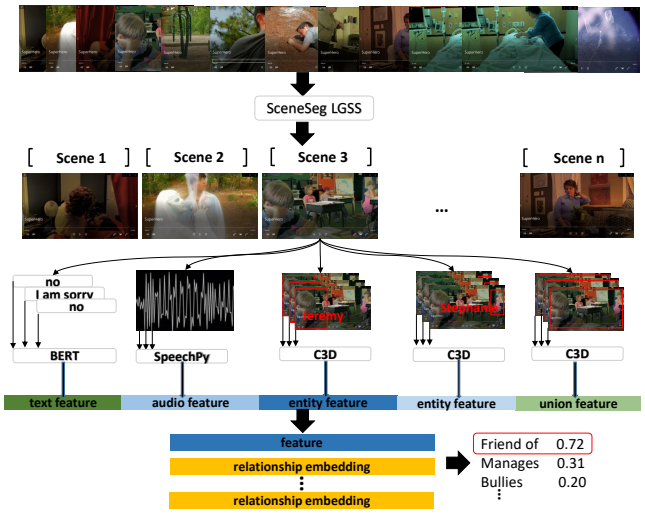

Figure 2: The pipeline of our method.

2 PRELIMINARY

Scene Segmentation. Scene segmentation is challenging because scenes in videos often contain abundant temporal structures and complex semantic information. Rao *et al.* [7] propose a local-to-global scene segmentation framework that can integrate multimodal information across three levels: clip, segmentation and movie. The framework can extract complex semantic information from the hierarchical time structure of a long movie and provide top-down guidance for scene segmentation.

Audio Feature and Text Feature. To extract semantic information from audio, MFCC and and LMFE are two important audio features that could be extracted by SpeechPy, a useful tool for speech processing and feature extraction. In addition to audio feature, feature from speech text also matters. Before feature extraction, speech text could be generated by some subtitles generating services such as autoSUB, Aliyun and Youtube. BERT [5] is a popular model in natural language processing and it can be used to extract text features of a word or a sentence.

Surf Feature Matching. Traditional template matching methods are easy to make mistakes if the template has compression distortion. Feature matching methods use scale invariant features like speed up robust features (SURF) [1] to detect local feature of an image.

Person Identification. The InsightFace model [4] integrates face detection and face alignment into a framework like the MTCNN model [11], and it is also able to match detected faces with face samples. For person identification by tracking, an object tracking model CenterTrack [12] also performs well on person tracking. This method represents each target as a point in the center of its bounding box, and then tracks the center point in time order.

Relationships Between Movie Characters. Detecting relationships between entities in the movies is central to our project. Kukleva *et al.* [6] work on the interactions and social relationships between characters in a movie. It should be noted their model is designed for a video clip with two characters, but it is not suitable for the DVU task.

3 OUR METHOD

Our method is based on scene segmentation because a scene is a crucial unit of a movie, which contains complex activities of actors in a physical environment [7]. After scene segmentation, we recognize the entities of the location and all persons in each scene. Then, we extract multimodal features of each scene, including visual features, audio features and text features. The visual features of one scene include visual features of each entity and those of the union areas of each two entities. These features are then concatenated and transformed to compute similarities with different relationships. In this way, we can build our entity-relationship graph by combining relationships in all scenes, which provides support for the DVU task.

3.1 Scene Segmentation

We choose SceneSeg LGSS [7] to divide the movie into separate scenes. Its framework is able to distill complex semantic information from hierarchical temporal structures over a long movie, providing top-down guidance for scene segmentation. SceneSeg LGSS is able to split a video into scenes based on multimodal features. We implement the scene segmentation with place features, image features and audio features respectively and take the union of them as final result.In this way, we can avoid mistaking many short scenes as a long duration scene.

3.2 Location Recognition

Since a scene is matched to one location in definition, we try to recognize the location of each scene based on the provided screenshots. We extract SURF features of the location screenshots and those of the scene frames, and then match them according to their SURF feature distance. We choose the locations whose screenshots can be matched to at least a frame and the number of matched feature points exceed the threshold as the candidate locations of the scene. To find the best match for a scene, we calculate the sum of the matched feature points of one location and choose the top one as the location of the scene.

3.3 Person Recognition

To recognize persons in a scene and record their trace, we combine person tracking and person recognition. We use CenterTrack [12], which detects the center point of the human body and predicts the movements according to the center point to do the tracking, to track all persons in a movie and then save the traces. We use InsightFace [4], which detects faces with MTCNN and then match them with the provided images of person entities by ArcFace [3], which puts forward additive angular margin loss for deep face recognition, to recognize all faces appeared in the movie. Finally, we combine the results of body tracking and face recognition by matching body with face(name) if their bounding box IOU exceeds the threshold.

As some of screenshots cannot be matched to faces detected by InsightFace, we match the entity recognized by surf matching and the body detected by CenterTrack if they are in the same frame and their bounding box IOU exceeds the threshold. Also, all the body bounding boxes with the same tracking id will be matched to the same entity like tracking an entity.

3.4 Multimodal Features

We use multi-modal features, including text features, audio features, video features and entity features, to predict the relationship of entities of each frame.

Text features We use the speech-to-text tools provided by autoSUB, Aliyun and Youtube to obtain the lines of a movie. Then, we match the lines to the scene according to the time they appear. We consider lines within a period of time with the number of words over fifteen as a sentence. If the number of words is less than fifteen (the largest number to make sure the maximum length of final sentences is smaller than the limit of the BERT model we used), we will spell them with the following words as a sentence until the number of words reaches fifteen. When encountering empty sentences or pausing for more than 0.5 seconds, we will directly take the previous words as a sentence. Finally, we use the BERT model to convert sentences to vectors as text features.

Audio features We extract MFCC and LMFE features by SpeechPy from the audio of the movie, and then calculate their first and second differential features respectively. Finally, the feature cubes of MFCC and LMFE are joint together to represent the audio features.

Entity features Entity bounding boxes can be tracked based on location recognition and person recognition. Based on the algorithm of average interval sampling from entities' traces in each scene, the visual features of entities in the corresponding scenes are generated from the C3D model.

Union features We can also compute the union bounding boxes of two entities and take an average sample in each scene as the input of C3D to obtain the union features.

Combine features to predict relationship All the features are then concatenated together and transformed into a feature whose dimension is the same as that of a text feature for relationship prediction between entities.

3.5 Training and Inference

For the absence of some relationships in the training set, we use zero shot learning during training. The relationship descriptions are used to generate features in the same way as text features are generated from subtitles. Cosine similarity of the feature representing a pair of entities and the relationship features are computed. The loss function is computed as follows:

$$L = (1 - cos(\beta, \gamma))^2 + \frac{\sum_{i \in U} (cos(\beta, \mu_i) + 1)^2}{n}, \quad (1)$$

where L denotes the total loss, β denotes the feature of pair; γ denotes the feature of the positive relationship; U denotes the set of negative relationships; μ_i denotes the feature of relationship i; n denotes the number of negative relationships, which equals 58 according to HLVU dataset.

During inference, the cosine similarity of each pair feature and each relationship feature is treated as the final score. We also add rules about the categories and names of subjects and objects.

3.6 Query Answering

Based on the final entity-relationship graph, we response to the three types of queries. (1) To fill in spaces in the graph, we sort the candidates in our entity-relationship graph according to scores generated by our method. (2) For the question answering task, we traverse all the choices and examine whether our graph is satisfied. If none of the choices can fit our graph, we choose a reasonable answer based on the types of entities and relationships. (3) We collect the paths between two entities by depth-first searching through the graph.

4 EXPERIMENTS

4.1 Dataset and Experimental Settings

All the experiments are conducted with i7-8086K 4.00GHz 12 cores CPU, 64GB memory and one TITAN V GPU, on the HLVU dataset [2].

The HLVU dataset contains 10 movies from public websites, about 11 hours in total.The dataset provides each movie in the development set with a manually annotated knowledge graph that contains entities and their relationships. The

Table 1: Experiments on different variants and a baseline, where T represents text feature, E represents entity feature, U represents union feature, A represents audio feature, C represent rule about category and N represents rule about name.

Method	R@50	R@100	R@θ
T+E/+C/+C+N	2.959/5.325/8.876	9.467/10.651/17.160	2.367/3.550/9.467
T+E+U/+C/+C+N	8.876/9.467/14.201	**10.651**/**11.834**/17.160	7.692/**8.284**/**11.834**
A+E/+C/+C+N	2.367/2.959/8.284	8.284/11.243/17.160	2.367/2.959/8.284
A+E+U/+C/+C+N	0.592/1.775/8.284	2.959/8.284/13.018	0.592/1.183/7.692
T+A+E/+C/+C+N	**9.467**/9.467/**14.793**	10.059/**11.834**/**17.751**	**8.284**/**8.284**/10.651
T+A+E+U/+C/+C+N	**9.467**/**10.059**/**14.793**	**10.651**/11.243/**17.751**	5.917/7.101/**11.834**
C+N	8.284	13.609	7.692

dataset also provides a set of image and/or video examples of different actors and entities including important locations, each with a name ID. The HLVU dataset provides three different types of queries: (1) Fill in the graph space, (2) Question Answering, (3) Relations between characters. Our method is acquired to answer these three types of queries.

To evaluate the performance of our method, the following metrics are designed for the three query types. For "Fill in the graph space", results will be treated as ranked list of result items per each unknown variable and the Reciprocal Rank score Eq (2) will be calculated per unknown variable and Mean Reciprocal Rank (MRR) per query.

$$MRR = \frac{1}{\lambda}\sum_{i=1}^{\lambda}\frac{1}{\mu_i}, \tag{2}$$

where λ denotes the number of unknown variables, μ_i denotes the rank of right answer of i_{th} unknown variable in the answer list. For "Question Answering", multiple choice questions will be provided for participants to answer based on the knowledge graph. The evaluation metric proposed for this query type is calculated by the number of correct answers/number of total questions. For "Relations between characters", we should evaluate whether each path is a valid path first, and report the recall, precision and F1 measures:

$$F1 = \frac{2pr}{p+r}, \tag{3}$$

where p represents precision and r represents recall.

In our experiments, we evaluate the performance of knowledge graphs generation using metric *Recall*@k, which is usually applied in visual relation detection. The metric *Recall*@k is computed by

$$Recall@k = \frac{TP_k}{TP_k + FN_k}, \tag{4}$$

where TP_k and FN_k denote the number of correct relations predicted and unpredicted in the top k confident relationship predictions, respectively. k is set to 50, 100 and θ (the number of ground truth relationships).

4.2 Component Analysis and Comparison with Baseline

We evaluate the effectiveness of different features and rules used in our method on the test set. The results are shown in Table 1.

We first design six variants to evaluate the effectiveness of different features. The first variant uses text and entity features; the second uses text, entity and union features; the third uses audio and entity features; the forth uses audio, entity and union features; the fifth uses text, audio and entity features; the sixth uses text, audio, entity and union features. From the statistics, we find that text, audio and visual features all contribute to good performance. Then, we add some rules based on the model using multimodal features. These rules are used to predict relationships between two entities according to their names and types. Experiments validate that adding these rules effectively improves the performance whenever using any features.

We consider the method that only uses rules about type and name as a baseline. The metric values of this baseline are in the last line in Table 1. From the comparison between the baseline and our method not using rules, we can see that the variant using text, audio, entity and union features performs better at *Recall*@50 and *Recall*@θ. We presume that the baseline reports better results for *Recall*@100 because the number of ground truth relationships is much less than 100 and rules about category and name tend to cover more relationships that are usual in common sense. However, when our method adds these rule, the final result tends to exceed that of the baseline.

5 CONCLUSIONS

In this paper, we proposed a multimodal feature fusion method to extract knowledge graphs of movies for deep video understanding based on the analysis of the relationships among entities in movies. The proposed method divides the given movie into separate scenes and extracts features to detect the relationships among entities in each scene. We evaluated our method on the HLVU dataset, and the experimental results validated the effectiveness of our method.

ACKNOWLEDGEMENT

This work is supported by Natural Science Foundation of Jiangsu Province (BK20191248), Science, Technology and Innovation Commission of Shenzhen Municipality (JCYJ20180307151516166), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. 2006. Surf: Speeded up robust features. In *European Conference on Computer Vision*. 404–417.

[2] Keith Curtis, George Awad, Shahzad Rajput, and Ian Soboroff. 2020. HLVU: A New Challenge to Test Deep Understanding of Movies the Way Humans do. In *International Conference on Multimedia Retrieval*. 355–361.

[3] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[4] Jiankang Deng, Jia Guo, Zhou Yuxiang, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. 2019. RetinaFace: Single-stage Dense Face Localisation in the Wild. *arXiv preprint arXiv:1905.00641* (2019).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Anna Kukleva, Makarand Tapaswi, and Ivan Laptev. 2020. Learning Interactions and Relationships between Movie Characters. In *IEEE Conference on Computer Vision and Pattern Recognition*. 9849–9858.

[7] Anyi Rao, Linning Xu, Yu Xiong, Guodong Xu, Qingqiu Huang, Bolei Zhou, and Dahua Lin. 2020. A Local-to-Global Approach to Multi-modal Movie Scene Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 10146–10155.

[8] Mrigank Rochan and Yang Wang. 2019. Video Summarization by Learning from Unpaired Data. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[9] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. 2017. Video Visual Relation Detection. In *ACM International Conference on Multimedia*. 1300–1308.

[10] Jinhui Tang, Xiangbo Shu, Rui Yan, and Liyan Zhang. 2019. Coherence Constrained Graph LSTM for Group Activity Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), 1–1.

[11] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters* (2016), 1499–1503.

[12] Xingyi Zhou, Vladlen Koltun, and Philipp Krhenbhl. 2020. Tracking Objects as Points. *arXiv preprint arXiv:2004.01177* (2020).