Reproducibility Companion Paper of "MMSF: A Multimodal Sentiment-Fused Method to Recognize Video Speaking Style"

Fan Yu
yf@smail.nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Beibei Zhang
zhangbb@smail.nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Yaqun Fang
fangyq@smail.nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Jia Bei
beijia@nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Tongwei Ren*
rentw@nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Jiyi Li
garfieldpigljy@gmail.com
Department of Computer Science and Engineering, University of Yamanashi
Kofu, Japan

Luca Rossetto
rossetto@ifi.uzh.ch
Department of Informatics, University of Zurich
Zurich, Switzerland

ABSTRACT

To support the replication of "MMSF: A Multimodal Sentiment-Fused Method to Recognize Video Speaking Style", which was presented at ICMR'23, this companion paper provides the details of the artifacts. Speaking style recognition is aimed at recognizing the styles of conversations, which provides a fine-grained description about talking. In the original paper, we proposed a novel multimodal sentiment-fused method, MMSF, which extracts and integrates visual, audio and textual features of videos and introduced sentiment in MMSF with cross-attention mechanism to enhance the video feature to recognize speaking styles. In this paper, we explain the details of the implement code and the dataset used for experiments.

CCS CONCEPTS

• **Computing methodologies → Computer vision**.

KEYWORDS

Speaking style recognition, multimodal analysis, sentiment analysis, long-form video understanding

ACM Reference Format:
Fan Yu, Beibei Zhang, Yaqun Fang, Jia Bei, Tongwei Ren, Jiyi Li, and Luca Rossetto. 2024. Reproducibility Companion Paper of "MMSF: A Multimodal Sentiment-Fused Method to Recognize Video Speaking Style". In *Proceedings of International Conference on Multimedia Retrieval (ICMR'24)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 ARTIFACTS DESCRIPTION

1.1 Introduction

Speaking style recognition (SSR) aims to identify the conversation styles among characters in videos, which conducts deep studies on human conversations and contributes to further understand human activities in long-form videos [1, 2]. For more accurate prediction of SSR, we propose MMSF, a multimodal sentiment-fused method, which introduces sentiment influence into multimodal pipeline to enhance video feature [8].

The artifacts for replication of MMSF include the source code of the model, the docker image with running dependencies, additional supporting softwares, model weights for feature extraction and model prediction, the extracted features of the dataset we used for model training and inference, and the final prediction results. The source code is available at https://github.com/njumagus/MMSF and the download links of other artifacts are provided in the README file. It's crucial to emphasize that we do not provide the raw videos in line with the LVU dataset [4] used in our original paper's experiments. Instead, we offer the YouTube IDs of the videos along with the downloading script. Researchers are expected to manually download the raw videos or utilize the provided downloading script.

[*] Corresponding author.

ICMR'24, June 10–13, 2024, Phuket, Thailand
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1.2 Source code

The source code includes three parts: data process, feature extraction and the main model. After downloading all the other artifacts and moving them into the corresponding folders of the source code, the outmost folders and files are organized as follows:

- **data**: saving dataset and the processed data.
- **data_process**: data processing code.
- **feat_extract**: feature extracting code.
- **MMSA_feat**: saving extracted features.
- **prediction**: main model code.
- **sources**: additional supporting softwares.
- **bert-base-uncased**: saving bert model weights.
- **README.md**: source code explanation.

Data process The source code for data process is under the "data_process" folder and the data are stored under the "data" folder. The code structure of the two folders are organized as follows:

- **data**
  - **LVU**: saving data of the LVU dataset.
  - **config.py**: working as the configuration file of data split.
- **data_process**
  - **audio_process.py**: splitting audio from videos.
  - **data_download.py**: organizing raw data of the LVU dataset and downloading the videos.
  - **video_process.py**: processing the raw videos to clean the data.
  - **subtitle_process.py**: generating text of the video subtitles.

Feature extraction The source code for feature extraction is under the "feat_extract" folder and the features are saved under the "MMSA_feat" folder [3]. Sentiment features are extracted by a MMSA and Self-MM [5, 6] model trained on MOSI dataset [7]. The code structure of the two folders are as follows:

- **feat_extract**
  - **MMSA**: supporting feature extracting in MMSA_senti_feat.py.
  - **MMSA-FET**: saving temporal data and logs.
  - **MMSA_feat_configs**: saving configuration files for feature extraction.
  - **MMSA_regression_checkpoints**: saving model weights for feature extraction.
  - **MSA_FET**: supporting feature extracting in MMSA_feat.py.
  - **MMSA_feat.py**: extracting video, audio and text features.
  - **MMSA_senti_feat.py**: extracting sentiment features.
- **MMSA_feat**
  - **LVU**: saving extracted features.

Main model The source code of the main model is under the "prediction" folder. The code structure of the main model is organized as follows:

- **prediction**
  - **checkpoints**: saving model weights.
  - **model**
    - **pretrained_models**
      - **self_mm-mosi_lvu_outputdim300_bertfinetune.pth**: pretrained self_mm model weights.
    - **utils**
      - **BertTextEncoder.py**: working as the text encoder for self_mm.
      - **model_base_extractor.py**: extracting features of sentiment.
      - **model_selfmm.py**:
      - **multihead_attention.py**: working as multi-head attention in Transformers.
      - **position_embedding.py**: working as position embedding in Transformers.
      - **transformer.py**: working as Transformer encoder.
    - **model_MMSA.py**: working as the main model.
  - **results**: saving results of model inference.
  - **config.py**: working as the configuration file.
  - **dataset_MMSA.py**: loading data for training and inference.
  - **evaluate.py**: evaluating inference results.
  - **run_mmsa.py**: working as the main running script.

2 EXPERIMENTS

2.1 Environment

Operating System All the experiments are conducted at Ubuntu 18.04 LTS with CPU E5-2680 v4, 128GB memory and 2TB free space. We only use one GTX4090 GPU with CUDA 11.8. It is worth noting that the GPU used in replication is different from that in the experiments of the original paper.

System Software Some softwares are required to be installed at system level: libopenblas-dev, git, libgtk2.0-dev, pkg-config, libavcodec-dev, libavformat-dev, libswscale-dev, python-dev, python-numpy, libtbb2, libtbb-dev, libjpeg-dev, libpng-dev, libtiff-dev, libdc1394-22-dev, libboost-all-dev, ffmpeg, OpenCV, Dlib, MSA_FET.

Python Package The python version we used is 3.8, and some python packages are required to be installed: pandas, yt-dlp, ffmpeg-python, deepspeech, scipy, torch, tqdm, opencv-python, scenedetect, python_speech_features, librosa, opensmile, transformers, mediapipe, gdown, easydict, pynvml, torchnet.

2.2 Dataset

Our experiments are conducted on LVU dataset [4]. There are nine tasks of LVU benchmark and SSR is one of them. Since some videos are not accessible from YouTube and some will encounter errors in feature extraction, the final dataset we used in our experiment contains 1,336 movie clips in total, comprising 935 training videos, 203 validation videos, and 198 test videos. In this task, each movie clip lasts one to three minutes and corresponds to one speaking style label. There are five speaking style categories, namely Explain, Confront, Discuss, Teach and Threaten. Given that the raw videos, originally downloaded from YouTube, are no longer publicly accessible, we will not offer the raw data. Instead, we provide the YouTube IDs of the videos along with the downloading script:

```
python data_process/data_download.py
```

The raw videos need to be pre-processed by:

```
python data_process/video_process.py
```

Table 1: Important parameters that can be customized.

Parameter	Description	Default Value
modal_list	The feature modalities used in MMSF.	[“video”,“audio”,“text”]
senti_modal_list	The modalities used for sentiment prediction.	[“video”,“audio”,“text”]
senti_feat_fusion_strategies	The strategies used for fusing different modalities and sentiment.	{“video”: “fusion”, “audio”: “fusion”, “text”: “fusion”}
early_fusion	The flag to control using early fusion or late fusion.	False
lr	Initial learning rate.	1e-5
batch_size	The number of videos of each batch.	16
max_epoch	The max number of epoch.	100

For text feature extraction, subtitles of the videos should be generated. We first split audio from the videos and then transfer the characters’ speech to text:

```
python data_process/audio_process.py
python data_process/subtitle_process.py
```

It is noteworthy that additional videos became inaccessible from YouTube during the reproducibility process, and we have listed these videos in the comments of “data/config.py”. Researchers may encounter different inaccessible videos when attempting to execute our code, and they have the option to update the video list in “data/config.py” based on the data accessibility at that time.

2.3 Feature Extraction

Before model training and inference, we extract multimodal features and sentiment features. First, a “label.csv” file is generated to organize data information:

```
python feat_extract/MMSA_feat.py --label
```

Then, audio, video and text features are extracted as follows:

```
python feat_extract/MMSA_feat.py --audio
python feat_extract/MMSA_feat.py --video
python feat_extract/MMSA_feat.py --text
```

Finally, sentiment features are extracted by:

```
python feat_extract/MMSA_senti_feat.py
```

2.4 Model Training and Inference

To train our MMSF model, run the following script and model weights will be saved in “prediction/checkpoints”:

```
python prediction/run_mmsf.py --train
```

To test the model and evaluate the results, run the following script and the results will be saved in “prediction/results”:

```
python prediction/run_mmsf.py --test
```

The model parameters defined in the “prediction/config.py” file can be adjusted by a custom configuration file, and some important parameters as well as their description are shown in Table 1. The parameters should be adjusted for different environments. It is worth noting that our replicated result of MMSF achieves 50.0% Top-1 Accuracy and 47.0% F1-score, which is a little bit higher than that in our original paper. We provided the download link of the model weight and the result file in the README file.

3 REPRODUCIBILITY EFFORTS

The last two co-authors of this paper are the reviewers of this companion paper.

Luca Rossetto ran the code natively, based on the provided documentation, was reasonably straight-forward, although not effortless. This was not due to the code in this work itself but due to some of the external tools it uses as dependencies for data preprocessing and feature extraction. Some of these tools have very specific system-level dependencies themselves, which might no longer be available in the required versions in recent operating systems and had therefore be compiled from source. The authors provide a docker container environment to mitigate this problem. Another challenge was that the dataset on which the results presented in the original paper are based is sourced from YouTube and a substantial fraction of the videos are no longer available. This required some adjustment in the configuration of the project, as the code could not handle missing input files. Adjusting the configuration resolved this issue easily.

Jiyi Li checked through the manuscript and the materials. He found some inconsistent information in the manuscript, the source code, the data, and the documents. Besides the corrections, he also gave some comments to improve the presentations of the manuscript and documents, so that other researchers would be easier to follow the instructions. On the aspect of running the source code, for the “Quick Start” in the GitHub repository, he obtained the consistent accuracy and f1-score by using the provided checkpoint. He didn’t run the feature extraction program because the processed data and features are provided. He personally thinks that it’s acceptable if the processed data is provided and can be utilized to execute the source code. Because the raw data are also provided, other researchers still have the option to use the code to generate features for custom usage. For training and inference, on the one hand, he successfully ran the program for training and inference and reach the reported results; on the other hand, the results are not stable in different trials. Regarding the unstable results, the authors suspect that they may be attributed to the relatively small scale of the LVU dataset, the dropout structure in our network, and differences in machines, environments and random seeds. They will endeavor to investigate further in their future work.

In conclusion, based on the comments from the reviewers, the authors improved the paper and the materials. We hope this reproducibility work would be helpful for other researchers to utilize this paper.

4 CONCLUSION

In this paper, we provided the details of the artifacts of the paper "MMSF: A Multimodal Sentiment-Fused Method to Recognize Video Speaking Style" for replication. The artifacts contain the dataset and the source code for experiments in the paper.

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China (62072232), the Fundamental Research Funds for the Central Universities (021714380026) and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] Edward Fish, Jon Weinbren, and Andrew Gilbert. 2022. Two-Stream Transformer Architecture for Long Form Video Understanding. In *British Machine Vision Conference*.

[2] Md Mohaiminul Islam and Gedas Bertasius. 2022. Long movie clip classification with state-space video models. In *European Conference on Computer Vision*. Springer, 87–104.

[3] Huisheng Mao, Ziqi Yuan, Hua Xu, Wenmeng Yu, Yihe Liu, and Kai Gao. 2022. M-sena: An integrated platform for multimodal sentiment analysis. *arXiv preprint arXiv:2203.12441* (2022).

[4] Chao-Yuan Wu and Philipp Krahenbuhl. 2021. Towards long-form video understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1884–1894.

[5] Wenmeng Yu, Hua Xu, Fanyang Meng, Yilin Zhu, Yixiao Ma, Jiele Wu, Jiyun Zou, and Kaicheng Yang. 2020. CH-SIMS: A Chinese multimodal sentiment analysis dataset with fine-grained annotation of modality. In *Annual Meeting of the Association for Computational Linguistics*. 3718–3727.

[6] Wenmeng Yu, Hua Xu, Ziqi Yuan, and Jiele Wu. 2021. Learning modality-specific representations with self-supervised multi-task learning for multimodal Sentiment Analysis. In *AAAI Conference on Artificial Intelligence*, Vol. 35. 10790–10797.

[7] Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis-Philippe Morency. 2016. Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos. *arXiv preprint arXiv:1606.06259* (2016).

[8] Beibei Zhang, Yaqun Fang, Fan Yu, Jia Bei, and Tongwei Ren. 2023. MMSF: A multimodal sentiment-fused method to recognize video speaking style. In *ACM International Conference on Multimedia Retrieval*. 289–297.