Automated Creation of Intra-Video Social Comments

Hao Wu

Supervisor: Prof. François Pitié
Prof. Gareth J. F. Jones
Prof. Séamus Lawless

Department of Electronic & Electrical Engineering
Trinity College Dublin

This dissertation is submitted for the degree of
Doctor of Philosophy

March 2023

To my aunt, Yan, for her eternal love and support.

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Hao Wu
March 2023

Acknowledgements

Pursuing a PhD is a prolonged journey, this is a path full of loneliness and periodical sense of frustration for a fragile mind. I cannot even imagine how it could be accomplished without the support from the people around me. First and foremost I would like to extend my most profound gratitude to my supervisors: Prof. François Pitié and Prof. Gareth J. F. Jones. It has been my great honour to be guided by François, who is a conscientious supervisor, an excellent researcher and a kind friend. He constantly offers constructive feedback and insightful suggestions on my work, at times I was greatly inspired by his zeal for research work during my unproductive periods. Gareth is one of the most intelligent men that I have ever interacted with, he is sharp and accurate in solving problems but is also exceedingly patient with me during each of our conversations. I would like to thank Gareth for his professional advice, infinite patience, and encouraging words during the course of my PhD. I would also like to express my sincere gratitude to my late supervisor, Prof. Seamus (Shay) Lawless, who unexpectedly passed away after fulfilling his life dream of scaling Mt. Everest in May 2019. Shay was a considerate supervisor that allowed me to pursue the doctoral degree following my personal interests, the danmu topic. I still remember all the affirmatives and encouragements he had offered to all my juvenile ideas, these supports meant a great deal to me when I was bewildered at work. He was an enlightening and charming human being, an example that I wish to emulate for a lifetime.

I also wish to thank Prof. Owen Conlan, Prof. Naomi Harte and Prof. Anil Kokaram for their insightful advice on my PhD transfer report. I am deeply indebted to Prof. Jiayuan Yu and Prof. Ruihai Dong for helping me in the early stage of my PhD.

I am very fortunate to have met many amazing colleagues at Trinity College Dublin and the ADAPT centre, I would like to thank Dr. Yu Xu, Dr. Anirban Chakraborty, Dr. Gary Munnelly, Dr. Brendan Spillane, Dr. Esraa Ali and Dr. Bilal YousufProf. Annalina Caputo for the great time that we spent together.

I am thankful to ADAPT Centre for Digital Content Technology, which is funded under the Science Foundation Ireland Research Centres Programme (Grant 13/RC/2106) and is co-funded by the European Regional Development Fund, to support this research.

I would like to give my special thanks to my best friends, Dr. Zihui Li, Dr. Jinghui Lu, Dr Sixun Ouyang, for the happiness and sorrow we shared. They made this journey less lonely for me. As a matter of fact, Zihui is the person who introduced me to the field of deep learning and talked me into pursuing this PhD. I want to thank her for her help and encouragement.

I must extend my sincere appreciation to my girlfriend, Shiqi Shen, for her immense love and support. Shiqi, my life is much brighter ever since I met you, and without you standing by my side, I wouldn't be able to finish this PhD.

Finally, I want to thank my family, including my mother, Ying, my sister and my brother in law, Yue and Junmiao, and my aunt Yan. I am deeply grateful to Yue and Junmiao for their constant support and care ever since I came to Ireland, you made me feel at home and my life much warmer in this foreign land. I am most grateful to my aunt, Yan, for her unconditional love in every aspect of my life. She supports and encourages me at every single crossroad of my life. Quitting a stable job to pursue a PhD was a tough decision to make in 2017, Yan made this process so much easier for me with her firm support and encouragement in this matter. Her love is truly selfless and always motivates me during my PhD.

Abstract

Live video comments, or “danmu”, are an emerging social feature on Asian online video platforms. These time-synchronous comments are overlaid on the video playback and uniquely enrich the viewing experience, engaging hundreds of millions of users in rich community discussions. The presence of danmu comments has become a determining factor for video popularity. Videos with fewer danmu are not likely to be placed at the top in a search result list or to be recommended, therefore they receive less attention from viewers, which, in return, stops them from being further commented. This is similar to the cold-start problem in recommender systems. To overcome this cold start problem, we propose to automatically generate new danmu for less commented videos. Most of the existing literature on automated danmu creation has so far focused on generating danmu comments at random locations in already densely commented videos. However, the real issue faced by content creators is that videos need many danmu comments to start attracting traffic. Also, it is easier in these cases to exploit the numerous nearby comments to generate new comments.

In this thesis, we study this video cold start problem and examine how new comments can be generated automatically on less-commented videos. We first propose to leverage the available information from all modalities, including video visual signals, audio soundtracks and linguistic inputs (previous comments), into a single Deep Learning Transformer architecture. We also show that, by training our network for different scenarios of danmu comment densities, ranging from the complete cold start scenario to the scenario where the video has already many comments, our method can outperform the state-of-the-art in all situations, even surpassing human comments in terms of fluency and relevancy to the original video content.

To further tackle this cold-start challenge, rather than generating comments at random places in a video timeline we propose to solve the problem of *where* to publish danmu comments, which is something that has not yet been addressed in the literature. As danmu comments tend to aggregate at particular highlights, we propose to predict these popular locations in the video timeline by building on the same core architecture of our comment generation network. Results show danmu density trends can be reliably predicted from bare videos, thus proving that we can also predict *where* to publish comments in a video.

Instead of separating the process of predicting the location and content of a danmu comment, we recognise that both tasks of comment generation and highlight prediction can be actually addressed within a unified framework, using the same input modalities and same core Transformer architecture,

but with two different decoders. This unified network is trained in a multi-task manner. Our results show that this multi-task approach consistently outperforms the single-task baselines.

Finally, the performance of our overall system is evaluated by human evaluators, measuring not only the quality of generated content but also the appropriateness of the recommended commenting locations. The evaluation results show that, when compared to human comments, our generated automated comments are more relevant to the source video and their timing tend to be more accurate than human comments.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Multimedia on the Internet is growing rapidly. Video traffic constitutes 80 percent of all IP traffic [1]. For YouTube [2] alone, a billion hours of videos are watched every day. This growth in multimedia content is occurring within an environment in which Web 2.0 makes it possible for users to interact and collaborate with each other through social media dialogues. In the 2010s, the features of Web 2.0 became ubiquitous on social media platforms such as YouTube, Twitch [3], TikTok [4], Facebook [5], Twitter [6], Instagram [7] and Bilibili [8].

All of these social media platforms enable user commenting features, which have largely promoted user interaction and also provided rich first-hand feedback for both content-creators and platforms.

The traditional mechanism of video commenting is within a dedicated comment section below or next to the video player component. An example of a YouTube video along with its comment section is shown in Figure 1.1. Some of the social platforms such as TikTok and Instagram focus on building video social-networking platforms on mobile devices. While users in the web-based video sharing platforms tend to upload long videos, distinctively, these mobile-oriented social media platforms encourage users to make a variety of short-form videos, from genres like dance, comedy, and sports, that have durations from 15 seconds to one minute (see Figure 1.2).

A feature of these traditional commentary systems is that user comments are typically isolated from the video itself and displayed statically. In contrast, live-streaming applications have developed real-time commenting systems, where users can engage with the video in a more interactive way (see Figure 1.4). As a leading game streaming website, Twitch offers such real-time communication between streamers and audiences as shown in Figure 1.3. Other streaming sites such as YouTube-

[1] https://www.cisco.comc/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html

[2] https://www.https://www.youtube.com

[3] https://www.twitch.com

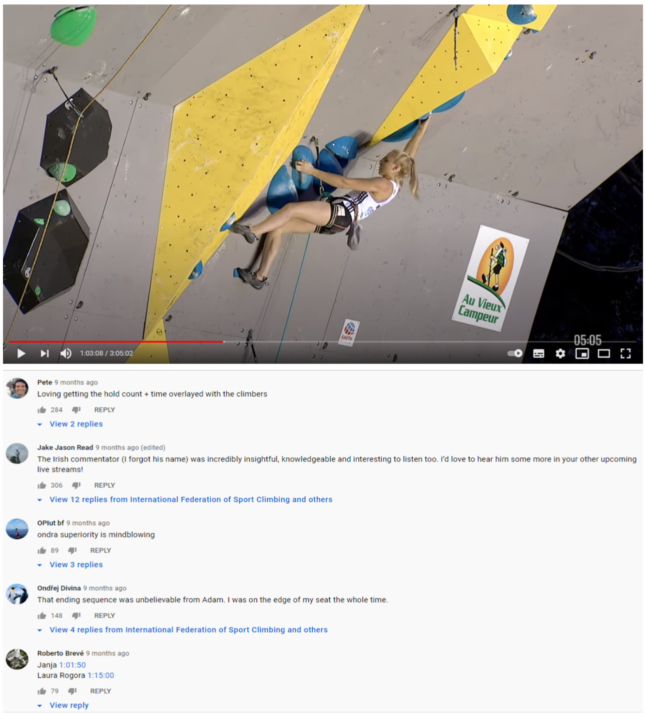
[4] https://www.tiktok.com

[5] https://www.facebook.com

[6] https://twitter.comhome

[7] https://www.instagram.com

[8] https://www.bilibili.com



Pete 9 months ago
Loving getting the hold count + time overlayed with the climbers
284 REPLY
View 2 replies

Jake Jason Read 9 months ago (edited)
The Irish commentator (I forgot his name) was incredibly insightful, knowledgeable and interesting to listen too. I'd love to hear him some more in your other upcoming live streams!
306 REPLY
View 12 replies from International Federation of Sport Climbing and others

OPlut bf 9 months ago
ondra superiority is mindblowing
89 REPLY
View 3 replies

Ondřej Divina 9 months ago
That ending sequence was unbelievable from Adam. I was on the edge of my seat the whole time.
148 REPLY
View 4 replies from International Federation of Sport Climbing and others

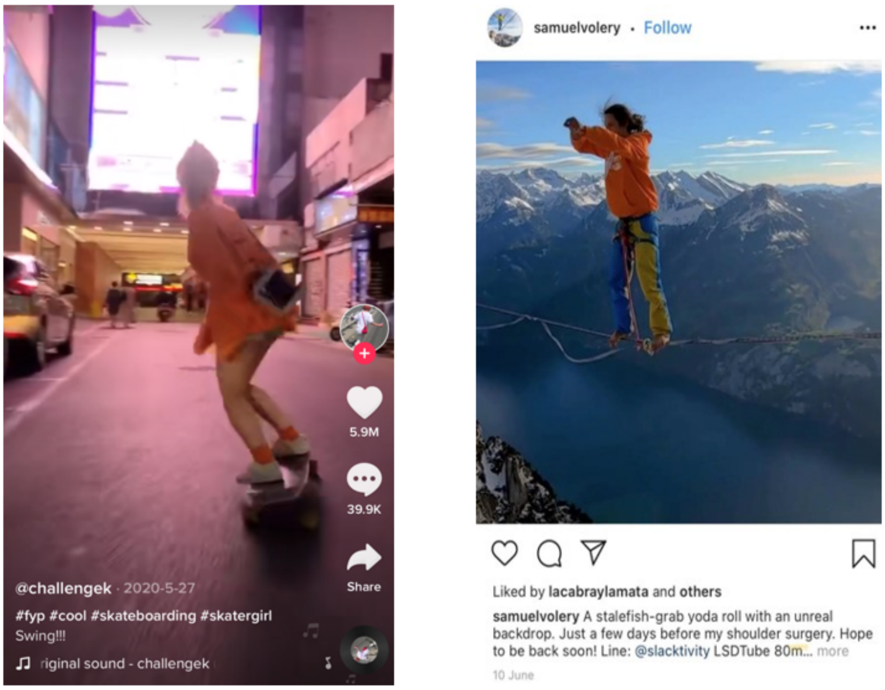
Roberto Brevé 9 months ago
Janja 1:01:50
Laura Rogora 1:15:00
79 REPLY
View reply

Fig. 1.1 The interface of a YouTube video with its comment section attached below.

Live [9] (see Figure 1.4) and Douyu [10] offer a similar commenting system, with the real-time user comments displayed on the right side of the video interface. This type of commenting system resembles a real-time online chat room in its form. The comments are synchronized with the streamed video and are thus more closely connected to the video content when compared to the traditional commenting systems. An obvious limitation is that the usage of real-time commenting is restricted to live-streaming content.

Recently, new types of video-sharing platforms such as Bilibili and Nicovideo [11] have emerged and are attracting increasing attention among young people in Asia. Bilibili has become the most popular video platform in China and has recorded 267 million monthly active users in the last quarter of 2021. It provides a unique user experience for its live commenting system as shown in Figure 1.5, at the bottom of the figure is a similar traditional comment section as on YouTube. Users can comment on a specific moment in the video. The comments are synchronized with the video timeline and embedded in the video frame (see the text on the upper side of the video frame in Figure 1.5). These comments were originally called "*danmaku*" in the Nicovideo Japanese platform and then "弹幕" in the Chinese Bilibili platform. In the rest of this thesis, we will refer to the Pinyin (a romanization system for Chinese characters) version: "danmu". Figure 1.6 shows an example of the video from Bilibili with a few danmu overlaid. The danmu system is different from the commenting system or

[9] https://www.youtube.com/live/about
[10] https://www.douyu.com
[11] https://www.nicovideo.jp



@challengek · 2020-5-27

#fyp #cool #skateboarding #skatergirl
Swing!!!

riginal sound - challengek

Liked by lacabraylamata and others

samuelvolery A stalefish-grab yoda roll with an unreal backdrop. Just a few days before my shoulder surgery. Hope to be back soon! Line: @slacktivity LSDTube 80m... more

10 June

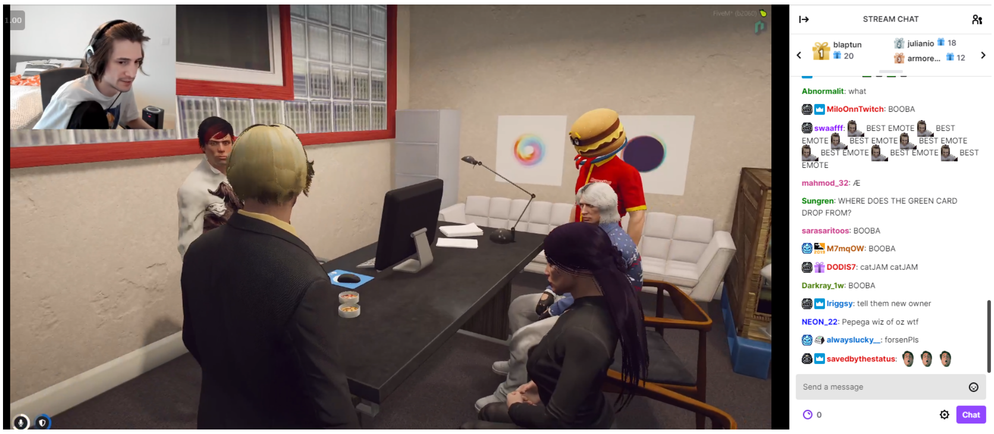
Fig. 1.2 Graphic on the left is the interface of a Tiktok video, the text below shows the uploader name, hashtags and background music. The graphic on the right shows the interface of an Instagram video, the uploader info is on the top and a short description of the video is placed at the bottom.

online streaming system on most video-sharing platforms. Since it provides a chat room experience in which users can watch and discuss together. Danmu comments frequently respond to the video content or surrounding conversations as shown in Fig 1.6, and provide dense opinion information which correlates to their specific locations in video timelines. With such rich and diverse interactive multi-modal signals, as well as being popular with online users, danmu videos are valuable for various research tasks such as video recommendation, sentiment analysis and automated comment generation.

We are particularly interested in these platforms as we think that by combining the state-of-the-art conversation response generation and multi-modal processing, we can develop new methods to automatically engage with or start new human dialogues on these video platforms.

1.1 Video Meta-data

The variety of commenting systems shows a range of user experiences, which indicate that video platforms nowadays are not limited to file sharing. There is a wealth of information around the videos which are encoded in meta-data, including textual meta-data and statistical meta-data (see Figure 1.7 for detailed examples of video meta-data). This extra information constitutes a major part of the marketing process in video platforms and is widely analysed for their recommender systems and search engines.



Abnormalit: what

MiloOnnTwitch: BOOBA

swaafff: BEST EMOTE BEST EMOTE BEST EMOTE BEST EMOTE BEST EMOTE BEST EMOTE BEST EMOTE

mahmod_32: Æ

Sungren: WHERE DOES THE GREEN CARD DROP FROM?

sarasaritoos: BOOBA

M7mqOW: BOOBA

DODIS7: catJAM catJAM

Darkray_1w: BOOBA

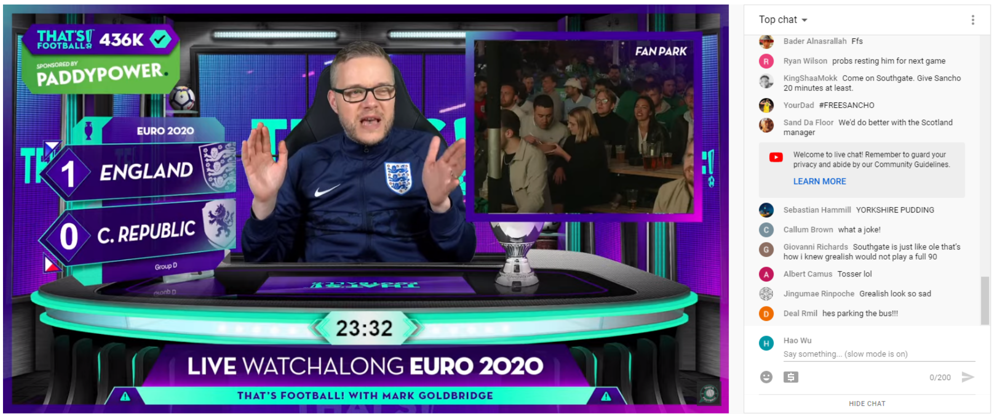
lriggsy: tell them new owner

NEON_22: Pepega wiz of oz wtf

alwayslucky__: forsenPls

savedbythestatus:

Fig. 1.3 The streaming interface of Twitch, on the right side is the real time commenting room for viewers.



Bader Alnasrallah Ffs

Ryan Wilson probs resting him for next game

KingShaaMokk Come on Southgate. Give Sancho 20 minutes at least.

YourDad #FREESANCHO

Sand Da Floor We'd do better with the Scotland manager

Welcome to live chat! Remember to guard your privacy and abide by our Community Guidelines.

LEARN MORE

Sebastian Hammill YORKSHIRE PUDDING

Callum Brown what a joke!

Giovanni Richards Southgate is just like ole that's how i knew grealish would not play a full 90

Albert Camus Tosser lol

Jingumae Rinpoche Grealish look so sad

Deal Rmil hes parking the bus!!!

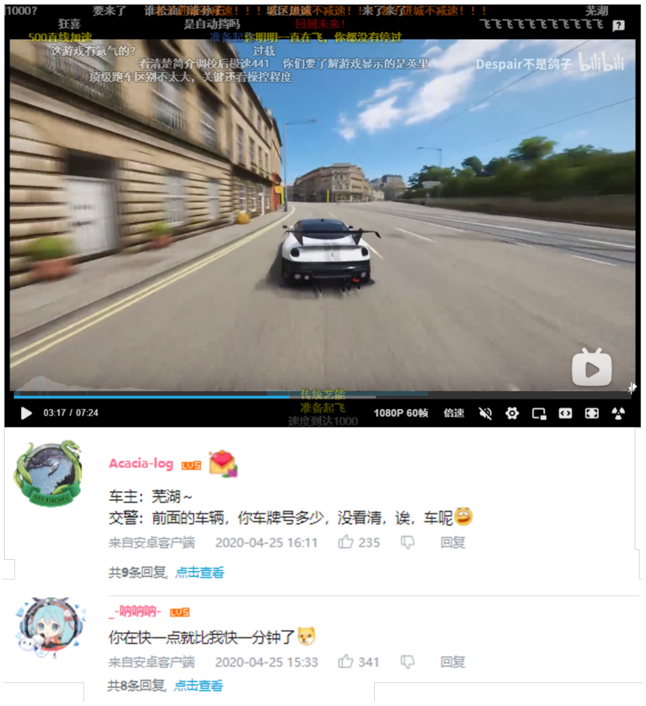
Fig. 1.4 The interface of a YouTube live channel, where the streaming video is on the left and the audience comments are displayed on the right.

Textual meta-data is usually provided by the uploader or auto-generated by the platforms. This typically includes short video descriptions, titles and hashtags. As a video uploader, deciding the most relevant keywords, title, or video description is a critical marketing decision, which will help to promote the video in the platform's recommendation systems.

Statistical meta-data such as the number of views, the number of likes/dislikes, grows during the lifespan of the video. These metrics can reflect the quality as well as the reputation of the video, thus becoming a non-trivial resource for analysing social videos.

From a video-sharing platform standpoint, meta-data information including user-created text meta-data and video statistics are the most important resources. Search engines and platform recommendation systems rely heavily on analysing this data (Davidson et al., 2010).

Apart from industrial exploitation, much research has also been conducted to analyse social videos through meta-data. For instance, Agarwal et al. (2017) propose to identify harassment in YouTube videos by generating video labels using text-based video meta-data features. Based on their



1000? 要来了 老司机进城不减速！！！ 老区进城不减速！！来了来了进城不减速！！！ 芜湖
狂喜 是自动挡吗 回到未来！ 飞飞飞飞飞飞飞飞飞飞飞
500直线加速 准备起飞 你明明一直在飞，你都没有停过
这游戏有氮气的？ 过载
看清楚简介调校后极速441 你们要了解游戏显示的是英里 Despair不是鸽子
顶级跑车区别不太大，关键还看操控程度

传统艺能
准备起飞
速度到达1000

03:17 / 07:24 1080P 60帧 倍速

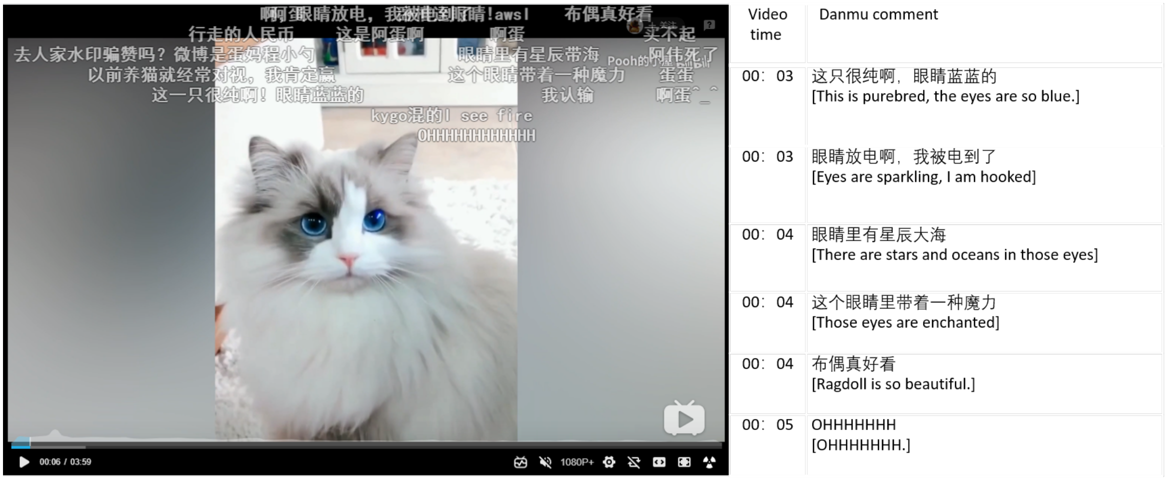
Acacia-log LV5
车主：芜湖~
交警：前面的车辆，你车牌号多少，没看清，诶，车呢
来自安卓客户端 2020-04-25 16:11 235 回复
共9条回复，点击查看

_-呐呐呐- LV5
你在快一点就比我快一分钟了
来自安卓客户端 2020-04-25 15:33 341 回复
共8条回复，点击查看

Fig. 1.5 The video interface of Bilibili, below are the comments selected from the comment section. On the upper side of the frame are the user comments, these comments are synced with the video timeline and displayed in the video frame.

investigation, video meta-data has been shown to be a highly informative resource for understanding the video as they achieve competitive results in the task of video labelling without analysing the actual video content. Another example is the work of Wu and Ito (2014), which analyses the statistical correlation between the popularity of a video on Bilibili and a series of video meta-data. They conclude that the popularity of a video on Bilibili is closely related to the number of coins (the virtual currency of Bilibili, users donate coins to support a video) collected. On the other hand, textual meta-data sometimes provides descriptive information of videos and can thus be utilized in multimedia data mining. Aradhye et al. (2009) use text meta-data as a weak supervision signal in annotating YouTube videos. This work transforms noisy video meta-data such as titles or user-uploaded descriptions into high-quality video labels, by searching for consistent correlations between raw video features and text meta-data.

Video meta-data helps us understand video content or the user community. Meanwhile, video comments serve as implicit user feedback and could be used to measure video popularity. Video platforms nowadays have adopted different commenting systems to bridge between video uploaders and viewers.



啊阿蛋眼睛放电，我被电到了睛!awsl 布偶真好看
行走的人民币 这是阿蛋啊 啊蛋 买不起
去人家水印骗赞吗？微博是蛋妈程小勺 眼睛里有星辰带海 阿伟死了
以前养猫就经常对视，我肯定赢 这个眼睛带着一种魔力 蛋蛋
这一只很纯啊！眼睛蓝蓝的 我认输 啊蛋^_^
kygo混的I see fire
OHHHHHHHHHHHH

Video time	Danmu comment
00：03	这只很纯啊，眼睛蓝蓝的 [This is purebred, the eyes are so blue.]
00：03	眼睛放电啊，我被电到了 [Eyes are sparkling, I am hooked]
00：04	眼睛里有星辰大海 [There are stars and oceans in those eyes]
00：04	这个眼睛里带着一种魔力 [Those eyes are enchanted]
00：04	布偶真好看 [Ragdoll is so beautiful.]
00：05	OHHHHHHH [OHHHHHHH.]

Fig. 1.6 The screenshot of a Bilibili video, some of the danmu comments on the topside of the frame are translated into English on the right.

1.2 Video Comments

Whereas title and description can be controlled by the content creator, user comments grow organically as viewers engage with the video. Both the video uploader and the platform have little control over this very important aspect of the video once it is placed on the platform. According to a survey from Schultes et al. (2013), 12 percent of YouTube users stated that they post comments regularly, while 34 percent stated that they read comments "often", and 53 percent agreed with the statement that they usually read the first two or three comments after watching a video. Since YouTube has over 1 billion users [12], we can deduce that the size of the active comment publishers is around 120 million. With such a huge community, comments are becoming one of the most important components of video-sharing platforms and of great value to all users of these platforms. On one hand, by reading and publishing comments, viewers perceive an added value in additional information, entertainment and social exchange. On the other hand, from the video uploader perspective, the community around the video comments provides first-hand feedback. Moreover, the comments around the video also relate to its popularity, a video with an active user community will be more likely to receive positive ratings and thus be recommended by the platform (Wu and Ito, 2014).

To understand how and why people use social media, Shao (2009) investigated the commenting behaviour in user-generated media. This study suggests that such user behaviour is mainly motivated by the following purposes: enhancing social connections and community development, achieving self-expression and self-actualization. When users consume content on video platforms, they also tend to participate through interacting with the content as well as with other users in the comment section for enhancing their social connections and virtual communities. From another perspective,

[12] https://merchdope.comyoutube-stats/

by creating and publishing comments they essentially produce their own content publicly to fulfil self-expression and self-actualization.

Social media comment sections provide valuable user feedback and enhance user engagements, however, they frequently attract social attention for the negative aspects of both their form and content. Comment sections of social media platforms have been known for frequent arguments and disagreements. According to the work of Duncan et al. (2020), the anonymity of online comment sections enables social media consumers to behave boldly and aggressively during discussions without fear of real-world accountability. The reason for this is also due to the fact that those with strongly held beliefs are more likely to comment and reply to others when the comments section is widely opposed to them, likewise, users tend to stay silent when their views are widely supported.

The social impressions of online comment sections are mostly negative, In 2013, an article in *Time* magazine (Grossman, 2013) criticized YouTube saying “The comments on YouTube make you weep for the future of humanity just for the spelling alone, never mind the obscenity and the naked hatred”. *The Guardian* (Owen, 2009) describe users’ comments on video platforms as: “Juvenile, aggressive, misspelled, sexist, homophobic, swinging from raging at the contents of a video to providing a pointlessly detailed description followed by a LOL, online video comments are a hotbed of infantile debate and unashamed ignorance with the occasional burst of wit shining through”.

In response to the phenomenon of the majority of user comments being perceived to be rude and offensive, many have been working on alleviating the impact of low-quality comments. In February 2017, Jigsaw [13], a Google-founded technology incubator, unveiled a tool based on artificial intelligence, called Perspective API, to identify toxic comments in online comment sections. Online video websites also take countermeasures, most video platforms have enabled report/block systems which help the platform to identify malicious comments. Bilibili provides an intelligence cloud filtering function, which would filter out low-quality comments based on user reports and automated detection tools.

In general, video comments reflect viewers’ perspectives on videos. Analysing this rich source of text information is a next step towards a better understanding of social videos and also online user interactions.

1.3 Research Applications

The scale of user-generated social videos has grown rapidly in recent years and has begun to attract enormous academic interest. With well-developed video processing and natural language processing (NLP) techniques, social video content along with its meta-data is analyzed in a wide range of research fields including sentiment analysis, video highlight detection and automated video commenting.

Video Comment Analysis With the prevalence of video sharing websites, user comments have become ubiquitous, and are often analyzed by the NLP community for various purposes. In 2019,

[13] https://jigsaw.google.com

Cunha et al. (2019) proposed to measure video reputation by analysing user comments. They applied a deep neural network (DNN) for the automatic classification of video comments into sentiment categories associated with users' opinions towards the video. Based on their results, the deep neural network largely improves task performance when compared with a statistical model, also it is shown in their work that the sentiments of user comments are closely related to video quality and popularity. Sentiment analysis of user comments can also contribute to the video searching process. Bhuiyan et al. (2017) proposed an NLP based sentiment analysis approach to help retrieve videos that are both relevant and popular. They show that it is possible to improve the video retrieval process by analysing user opinion in video comments.

Automated Video Commenting Automated commenting has become a hot research topic in recent years, this aims to generate human-like comments in a social media context. Rather than generating traditional comments, more effort has been focused on investigating live comments. Jumneanbun et al. (2020) generates rap-style video comments in live steaming to enhance audience experience. Ma et al. (2019) propose to use textual and visual contexts in the task of automated danmu commenting.

Highlight Detection Highlight detection is about finding the most interesting moments of a video, the data source of this task is usually from sports video (Min et al., 2014, Xiong et al., 2005, Assfalg et al., 2002) or user-generated social video (Xiong et al., 2019, Jiao et al., 2018, Ting et al., 2016, Qing and Chaomei, 2017, Zheng et al., 2020, Wu et al., 2021c).

For instance, Zhao et al. (2017) extract the best image thumbnail from YouTube videos through the analysis of both the video content and the textual video meta-data. As danmu comments to some extent reflect general user attention over the video, danmu video highlights can be identified by looking at the most commented locations in the video timeline. This was first explored by Zheng et al. (2020), who propose to predict video highlights by analysing danmu video content.

1.4 The Video Cold Start Problem

The number of danmu comments implicitly reflects the popularity of a video and also relates to the viewing experience. Videos with many danmu stand a higher chance of being recommended or searched and naturally attract more viewers. On the other hand, videos with fewer danmu are not likely to be placed at the top in a search result list or to be recommended. Moreover, many viewers will simply ignore videos with fewer danmu. After all, part of the fun of watching danmu based videos is to watch others discussing and to interact with other viewers using danmu.

The creation of new danmu to enrich videos has the potential to improve the viewing experience of viewers and to help attract more viewers. This is similar to the "cold start problem" in recommender systems (Lam et al., 2008). Until a video is annotated with a number of danmu it generally receives little attention from viewers, and will thus not become annotated with danmu. To overcome this cold start problem we propose to automatically generate new danmu for videos with no danmu or only a few danmu. For both a human and an automated commenting system, it is always harder to commence a topic or discussion than continue one. The lack of preexisting user comments leads to the first challenge as user danmu comments provide valuable conversational linguistic input for a comment generation system. On the other hand, the most popular existing conversation automatic comment generation approach tends to generate highly generic responses such as “I don’t know” or “Sounds good” (Li et al., 2016). However, the generated danmu comments need to be meaningful and engage other users to participate in the conversation by publishing their own danmu. In order to systematically overcome the cold start problem in danmu video commenting, we form our initial thoughts into research questions.

1.5 Research Questions

We propose that, analysing the video content and textual meta-data information can bring enough external knowledge for us to generate meaningful danmu. Indeed, most danmu conversations are closely related to the actual video content (Figure 1.6 is a good demonstration of this) and we believe that recent advances in computer vision and scene video understanding can help us recognise the crucial visual information from a danmu video (*e.g.* the cat with blue eyes) in order to automatically create meaningful and interactive comments on videos. Our aim is thus to explore the following research questions:

RQ1: Can we automatically create meaningful comments for less-commented or even uncommented videos?

RQ2: Can we automatically identify appropriate locations in video timelines to insert comments?

RQ3: How do our automatically created comments compare to human comments for the same videos?

1.6 Scientific Contributions

The major contributions of the thesis are as follows:

- We are the first to investigate the cold video problem for automated creation of danmu comments which enables us to create comments for freshly uploaded videos.
- We expand a publicly available danmu video dataset Ma et al. (2019) by doubling its size and enriching multi-modal features from video-embedded subtitles.
- We propose to add the audio soundtrack and video subtitles in addition to textual and visual input, we investigate the contributions of various modalities in the danmu generation task.
- We proposed a multi-density training regime, built around a multi-modal Transformer network, that can operate with videos of any number of preexisting comments.

- We use the danmu comment density as an indicator to classify danmu video highlights and address the problem of *where* to publish comments in a video.
- We propose to solve tasks of comment generation and highlight prediction at once, by leveraging a unified multi-task framework with a shared video encoder. Through extensive experiments, we prove that both tasks can contribute to each other during training and our unified approach yield better performance on both tasks.
- We design a human evaluation plan that can systematically measure the performance of a danmu commenting system through various dimensions.

1.7 Thesis Organisation

This thesis is organized as follows:

Chapter 2: Literature Review We conduct a literature review from a wide range of research fields that are related to our research goal, including the topics of natural language generation, multimedia processing and social media content analysis. The literature review provides us with valuable instructions and resources in understanding the state of the art technologies required to build an automated danmu commenting system.

Chapter 3: Danmu Comments Dataset This chapter elaborates the anatomy of danmu videos from a research point of view, covering their characteristics, functionalities and available data streams. We also present our danmu video collection which expands a public available dataset (Ma et al., 2019) with respect to both its size and dimensions. The expanded danmu video collection provides sufficient data resources to conduct the experiments in our later studies.

Chapter 4: OpenNMT-LiveBot: A Baseline Comment Generation Platform The Livebot (Ma et al., 2019) system serves as a starting point for our investigations towards our research objective by its well-established task definition, evaluation plan and baseline implementation. In Chapter 4, we first examine the issues relating to both the codebase and the dataset of the danmu comment generation framework released by LiveBot. Following this we report updated benchmark figures and re-implement the danmu commenting pipeline system in order to provide a reproducible implementation for our later research on the danmu commenting task.

Chapter 5: The Cold Video Problem in Danmu Comments Generation After building the pipeline system for the automated danmu commenting task, we present our exploration towards our first research question in Chapter 5. To consider the cold start scenarios of danmu videos, we reconstruct snapshots of a video's early commenting lifetime by masking a percentage of existing comments by their publication date. We then approach RQ1 by incorporating different levels of cold start scenarios in the training of the danmu commenting system.

Chapter 6: Danmu Video Highlight Prediction In this chapter, we describe our solution to our second research question of identifying appropriate locations for automated comments, where we investigate the distribution pattern of danmu comments over video timelines, and use the danmu comment density as an indicator to classify danmu video highlights. We further propose to predict danmu video highlights in an end-to-end manner by re-purposing the video encoder from the danmu commenting task.

Chapter 7: A Unified Multi-Task Approach We present our unified approach that jointly solves the first and the second research questions. This is implemented by considering a multi-task framework that trains both the comment generation module and the highlight detection module in a parallel way. We show in this chapter that this unified framework is not only more technically elegant, but also enables performance improvements over the two tasks.

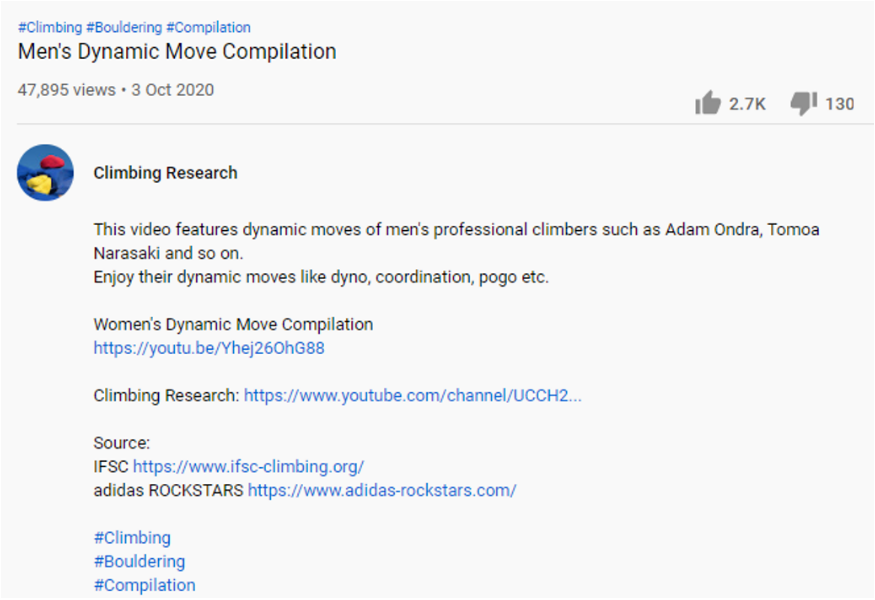
Chapter 8: System Evaluation In Chapter 8, we aim to solve the last research question by proposing a danmu commenting human evaluation plan with the consideration of our objectives for predicting both the content and the insert locations of the comments. We deploy a real-time user interface to simulate the original user experience of a danmu video website and recruit three Mandarin speakers with frequent danmu video viewing experiences as annotators.

Chapter 9: Conclusions and Future Research Directions Finally, we conclude the thesis by reviewing each individual research question and summarising the findings that arise from our investigations in solving them. We also present directions for potential future research extending our current work on danmu video commenting.

1.8 Publications

- H. Wu, F. Pitié, and G. J. F. Jones. Response to livebot: Generating live video comments based on visual and textual contexts. *arXiv*, 2020. (see Chapter 4)
- H. Wu, F. Pitié, and G. J. F. Jones. Cold start problem for automated live video comments. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 54–62, 2021b. (see Chapter 5)
- H. Wu, F. Pitié, and G. J. F. Jones. Investigating automated mechanisms for multi-modal prediction of user online-video commenting behaviour. In *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2021b. (see Chapter 6)

- H. Wu, G. J. F. Jones, and F. Pitie. Knowing where and what to write in automated live video comments: A unified multi-task approach. In *Proceedings of the 2021 International Conference on Multimodal Interaction (ICMI)*, pages 619–627, 2021a. (see Chapter 7)



#Climbing #Bouldering #Compilation

Men's Dynamic Move Compilation

47,895 views • 3 Oct 2020

Climbing Research

This video features dynamic moves of men's professional climbers such as Adam Ondra, Tomoa Narasaki and so on.
Enjoy their dynamic moves like dyno, coordination, pogo etc.

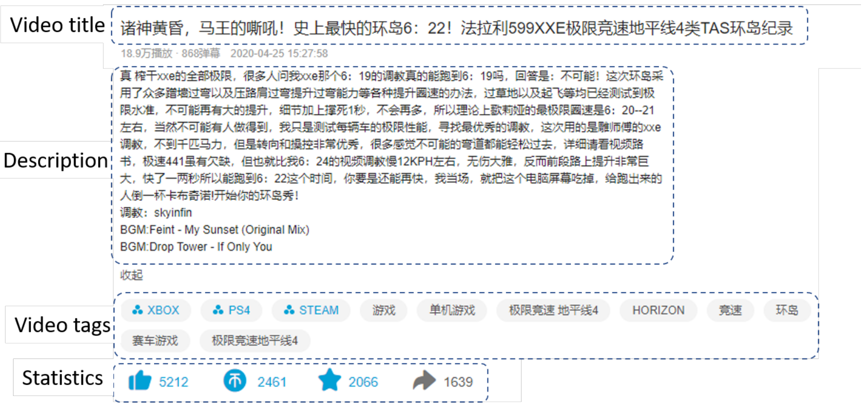
Women's Dynamic Move Compilation
https://youtu.be/Yhej26OhG88

Climbing Research: https://www.youtube.com/channel/UCCH2...

Source:
IFSC https://www.ifsc-climbing.org/
adidas ROCKSTARS https://www.adidas-rockstars.com/

#Climbing
#Bouldering
#Compilation

(a) YouTube video meta-data.



诸神黄昏，马王的嘶吼！史上最快的环岛6：22！法拉利599XXE极限竞速地平线4类TAS环岛纪录

18.9万播放 · 868弹幕 2020-04-25 15:27:58

真 榨干xxe的全部极限，很多人问我xxe那个6：19的调教真的能跑到6：19吗，回答是：不可能！这次环岛采用了众多蹭墙过弯以及压路肩过弯提升过弯能力等各种提升圈速的办法，过草地以及起飞等均已经测试到极限水准，不可能再有大的提升，细节加上撑死1秒，不会再多，所以理论上歌莉娅的最极限圈速是6：20--21左右，当然不可能有人做得到，我只是测试每辆车的极限性能，寻找最优秀的调教，这次用的是雕师傅的xxe调教，不到千匹马力，但是转向和操控非常优秀，很多感觉不可能的弯道都能轻松过去，详细请看视频路书，极速441虽有欠缺，但也就比我6：24的视频调教慢12KPH左右，无伤大雅，反而前段路上提升非常巨大，快了一两秒所以能跑到6：22这个时间，你要是还能再快，我当场，就把这个电脑屏幕吃掉，给跑出来的人倒一杯卡布奇诺!开始你的环岛秀！
调教：skyinfin
BGM:Feint - My Sunset (Original Mix)
BGM:Drop Tower - If Only You

XBOX PS4 STEAM 游戏 单机游戏 极限竞速 地平线4 HORIZON 竞速 环岛 赛车游戏 极限竞速地平线4

5212 2461 2066 1639

(b) Bilibili video meta-data.

Fig. 1.7 The meta-data of a video from both YouTube and Bilibili. Textual meta-data includes video title, video tags and video description. Other video statistics including the number of views, the number of likes and dislikes are also shown in the interface, videos of Bilibili have additional metrics including the number of coins, the number of bookmarks and the number of forwards

Chapter 2

Literature Review

The research question of generating new danmu comments requires a comprehensive analysis of various resources around danmu videos. Danmu videos are generally associated with a wide range of textual data, including user comments, video titles, video descriptions and embedded subtitles, and multi-media signals sourced from visual frames and audio soundtracks. To exploit these contextual resources in our task, we need to apply Natural Language Processing (NLP), Natural Language Generation (NLG) and Multi-modal Fusion techniques.

In this chapter, we review related existing literature in NLP, NLG and Multi-modal Fusion. Research work directly addressing danmu analysis and generation is reviewed and discussed in Chapters 4 and 5.

2.1 Natural Language Processing

With the goal of creating new danmu video comments, we are particularly interested in user danmu comments and video subtitles. We notice that, since danmu comments are always displayed with the video, the authors of danmu comments are not only reacting to the video content but also interacting with each other via their danmu comments and even engaging in discussions. The intrinsic connections between danmu comments are critical in creating new ones and thus need to be explored. We also observe that danmu comments frequently respond to speech in the video (see Figure. 2.1). These observations motivate us to process and utilize textual meta-data (existing user comments and video subtitles) in the analysis of danmu videos. Therefore, we first look at the research topics in the field of NLP.

2.1.1 Text Representations

Representing human language in a computer-readable format is a popular topic in NLP. This process is usually implemented by converting textual data into fixed-length vectors. The most straightforward representation is the one-hot vector, a $1 \times N$ matrix (vector) is used to identify each word in a

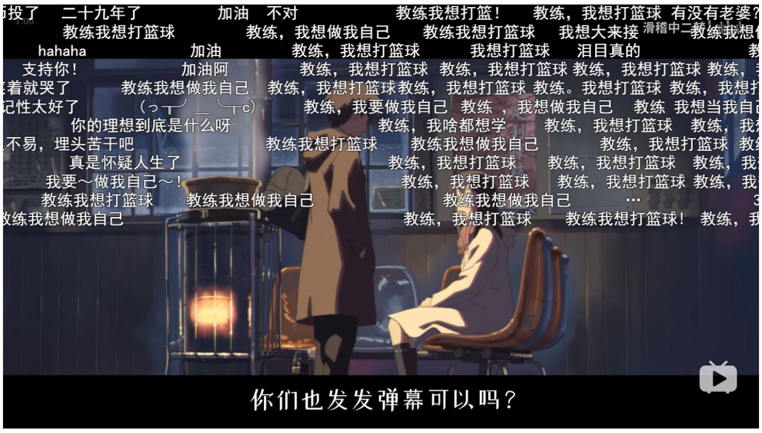

投了 二十九年了 加油 不对 教练我想打篮！ 教练，我想打篮球 有没有老婆？
教练我想打篮球 教练，我想做我自己 教练我想打篮球 我想大来接 滑稽中二 教练我想
hahaha 加油 教练，我想打篮球 我想打篮球 泪目真的 教
支持你！ 加油阿 教练，我想打篮球 教练，我想打篮球 教练，我想打篮球 教练，
着就哭了 教练我想做我自己 教练，我想打篮球教练，我想打篮球 教练。我想打篮球 教练，我想打
记性太好了 (っ╥╯﹏╰╥c) 教练，我要做我自己 教练 ，我想做我自己 教练 我想当我自
你的理想到底是什么呀 教练，我啥都想学 教练，我想打篮球 教练，我
不易，埋头苦干吧 教练我想打篮球 教练我想做我自己 教练，我想打篮球 教
真是怀疑人生了 教练，我想打篮球 教练，我想打篮球 教练，我
我要～做我自己～！ 教练，我想打篮球 教练，我想打篮球 教练，我
教练我想打篮球 教练我想做我自己 教练我想做我自己 … 3
教练我想做我自己 教练，我想打篮球 教练我想打篮球！ 教练，我
你们也发发弹幕可以吗？

Fig. 2.1 A video frame from bilibili.com with damnu comments overlaid. The subtitle says: “could you publish some danmu?”, and viewers are responding with a damnu burst.

vocabulary. The vector entries are 1 for the index of that word and 0 everywhere else. Similarly, one-hot encoding can be used at a character level, where N is the number of characters available. This kind of representation forms the basis of neural language models.

One limitation of this representation is that the length of a one-hot vector is always equal to the size of the vocabulary. This is not a problem at a character level encoding as the number of characters is usually quite small (one-hot character encoding is for instance widely used in building language models (Peng et al., 2003, Kim et al., 2016)). This is, however, a problem for word encoding as there are potentially several hundred thousand tokens in a language such as English or Chinese. Also, to deal with rare words that are outside of the fixed size vocabulary (out-of-vocabulary words), researchers usually use a single “UNK” symbol. This comes at the cost of a loss of information as these rare words could potentially be important (*e.g.* names, geographical landmarks, etc) in interpreting a text.

A compromise between character and word level encoding is proposed by Sennrich et al. (2016). They go below the word level by breaking words into smaller pieces. Representing words using sub-word units can significantly reduce the vocabulary size while resolving the problem of out-of-vocabulary words as any word can be generated using different combinations of word-pieces.

Besides the vocabulary size issue, one-hot encoding does not apply to common linear algebra computation (*e.g.* the cosine similarity of any 2 words will always be 0). This weakness makes further analysis difficult. Efforts like Word2Vec (Mikolov et al., 2013b), Glove (Pennington et al., 2014), Elmo (Peters et al., 2018), FastText (Joulin et al., 2017) etc. try to address this by finding a continuous latent word space where the inner product between word vectors indicate a degree of semantic relationship. For instance, Word2Vec attempts to relate the inner product of word vectors to the co-occurrence frequency of these words in a large corpus. These word embedding techniques induce interesting semantic characteristics and analogies. One famous analogy found across these techniques is that the word vectors for `man`, `woman`, `king` and `queen` are related as follows:

$$v_{\text{man}} - v_{\text{woman}} + v_{\text{king}} \approx v_{\text{queen}} \tag{2.1}$$

Text representation can go beyond word level. To learn highly generic sentence representations, Kiros et al. (2015) propose Skip-Thought Vectors, which apply skip-gram model (Mikolov et al., 2013a) at the sentence level, instead of using a word to predict its surrounding context, they encode a sentence to predict the sentences around it. Above the sentence level, some tasks like text classification or clustering require text representation at the document level. Le and Mikolov (2014) propose Paragraph Vector that learns continuous distributed vector representations for a paragraph or even a document.

The most popular embedding today is BERT (Devlin et al., 2019) (Bidirectional Encoder Representations from Transformers). The approach proposes pre-trained text representations, ranging from not only word embedding but also sentence embedding and whole paragraph embedding. The BERT model has proved to be very successful at NLP tasks. An advantage of BERT is that the proposed pre-trained embeddings can easily be fine-tuned to any specific vocabulary/corpus with a simple additional layer. Based on BERT, a series of large-scale language models (Yang et al., 2019, Radford et al., 2019, Dai et al., 2019) have been proposed and keep breaking the records on NLP tasks like question answering (Yang et al., 2019, Radford et al., 2019), machine translation (Radford et al., 2019), reading comprehension (Yang et al., 2019, Radford et al., 2019), and text summarisation (Yang et al., 2019, Radford et al., 2019).

Chinese NLP Returning to our research question, since the danmu data we are dealing with are Chinese language content, it is worth considering the difference between Chinese NLP and English NLP. Unlike English text where words are delimited by white spaces, in Chinese text there are no spaces between words and sentences are represented as strings of Chinese characters without explicit delimiters. Therefore, the first and the most important step in Chinese NLP tasks is to identify words from character sequences. Various methods have been proposed to address word segmentation, including statistical-based approaches (Peng and Schuurmans, 2001, Ge et al., 1999) and deep learning-based approaches (Cai and Zhao, 2016, Chen et al., 2015b,a, Cai et al., 2017).

In order to accurately segment Chinese words, one needs a proper way to handle out-of-vocabulary words (rare words that appear in the test set but not in the vocabulary of the training set.) and resolve word ambiguity problems since a Chinese character can occur in different word-internal positions with different interpretation (Xue, 2003). Inappropriate word segmentation can lead to the incorrect interpretation of the text source. Figure 2.2 is an example of Chinese word segmentation, the original Chinese sentence can be interpreted in two ways based on segmentation strategy. While it is true that both of the translations are grammatically correct independently, finding the contextually correct segmentation is important and a challenge. Sun and Tsou (1995) propose to solve word ambiguity in Chinese word segmentation using the maximal matching algorithm which incorporates both bi-gram statistics and linguistic rules for ambiguity resolution, according to the results, their method can successfully resolve most of the ambiguities in practical applications such as post-processing of OCR or speech recognition. Recently, sub-word units (Sennrich et al., 2016) has been used to shift the burden of word segmentation component in Chinese NLP, as a wrongly-segmented word may still

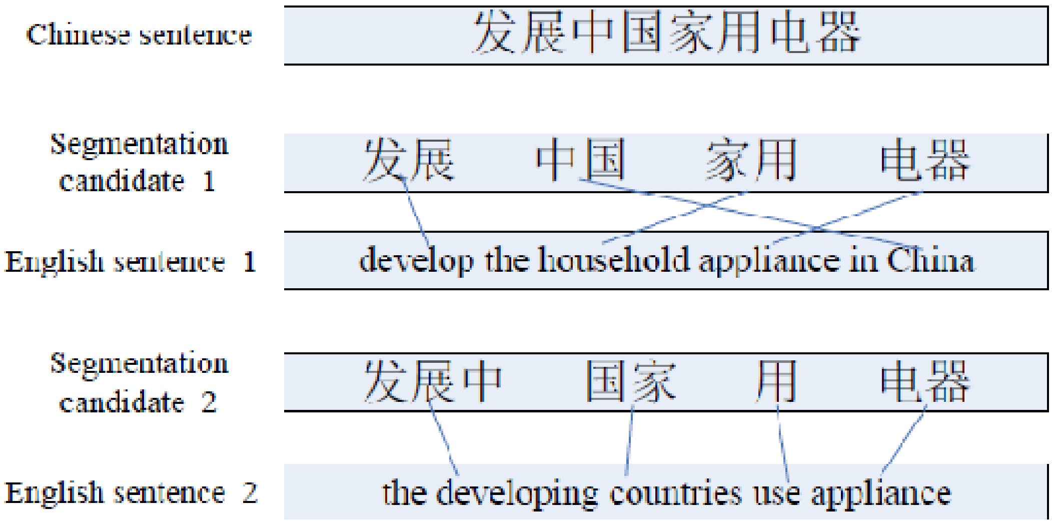


Fig. 2.2 A illustration of two different segmentations of a same given Chinese sentence.

be broken into smaller units. Wang et al. (2017) carry out an empirical study that compared various granularities (sentence, word, sub-word units and character level) in Chinese-to-English translation tasks and showed that sub-word units to be the most effective Chinese text representation in the experiments. Du and Way (2017) provide a new angle by converting Chinese characters into their Pinyin form (a romanization system for Chinese characters) and apply sub-word method on top of Pinyin, they also included denoting tones of Pinyin to enrich the text representation.

2.1.2 Natural Language Generation

NLG is a sub-field of NLP that is concerned with producing meaningful natural language from various contexts. This field is considered one of the most challenging tasks in NLP and has been studied since the 1960s. Researchers from the NLP community have been working on generating human language for years and have developed many applications in the fields of machine translation (Sennrich et al., 2016), conversation response generation (Xing et al., 2017), image caption (Yu et al., 2016) and text summarization (Li et al., 2017).

Rule-based Systems Some of the early successes in the field of NLG were mainly rule-based systems. Weizenbaum (1966) developed Eliza in 1966, which enables human-computer conversation by applying pattern matching and substitution methodologies. In 1975, Colby introduced PARRY (Colby, 2013), the dialogue system implements a rule-based model to simulate the behaviour of a paranoid schizophrenic. These approaches are able to produce human-like responses and are still the most widely used in practical applications (Gkatzia, 2016). This is partly because rule-based approaches are robust and can produce high-quality output given sufficient development time and cost. In addition, the outputs of these approaches are fully under control, making them generally accurate in their representation of the data. However, the performance of the rule-based systems is heavily reliant on manually written templates and pre-defined rules. The rule-based nature of these systems also makes their learning capacity very limited, and they are unable to be easily adapted/trained in new corpus and domains. As an alternative, statistical methods were introduced with a goal of providing a solution to these shortcomings.

Statistical Approaches Text generation systems shifted from traditional rule-based approaches to statistical approaches when researchers started to exploit patterns in the corpus and built models to make predictions of words given the statistical context. Until recently, similar statistical approaches were widely adopted in the field of machine translation. In 1992, Brown et al. (1992) proposed a statistical approach to machine translation from French to English, they introduced an algorithm for estimating the probabilities of a given English word being translated into any French word. The estimated probabilities are used in a statistical model to align words in the translation process. Statistical models also showed promising results in other text generation tasks such as summarisation. Barzilay and Lee (2004) proposed the Probabilistic Content Model (PCM) which leverages Hidden Markov Models for text summarisation, the proposed method captures dependencies on topic selection and organization for texts in a particular domain and yields substantial improvement over older methods.

In recent years, deep learning based approaches have been greatly studied to mitigate the issues faced by statistical language models such as word alignment (Yang et al., 2013), transition rule probability estimation (Devlin et al., 2014) and phase ordering (Xiong et al., 2006).

Deep Learning Methods With the large amount of corpora and significant computational resources becoming available in the last decade, deep learning models have emerged which show significant improvements in NLG effectiveness. Mikolov et al. (2010) argued that there had not been any significant progress in using statistical approaches to model language. This observation motivated his experimentation on using recurrent neural networks (RNN). This work laid a solid foundation for neural networks becoming a model of choice for modelling sequential data like text. One landmark example of utilizing RNN structure is the sequence-to-sequence model proposed in 2014 by Sutskever et al. (2014). This general end-to-end sequence learning approach uses a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimension, and then another deep LSTM to decode the target sequence from the vector. This work applies the soft attention mechanism (Bahdanau et al., 2015) to help the decoder focus on the relevant input tokens in each decoding step (see Figure 2.3 for detail). The model achieved the best result in the WMT-14 English to French translation task in 2014 and is widely used in many NLG tasks such as neural machine translation (Sennrich et al., 2016), text summarization (Li et al., 2017), video caption (Venugopalan et al., 2015) and dialogue response generation (Graves, 2013).

The Transformer Architecture In 2017, Google proposed the Transformer (Vaswani et al., 2017), an innovative sequence modelling architecture, as an alternative to the LSTM structure. The Trans-

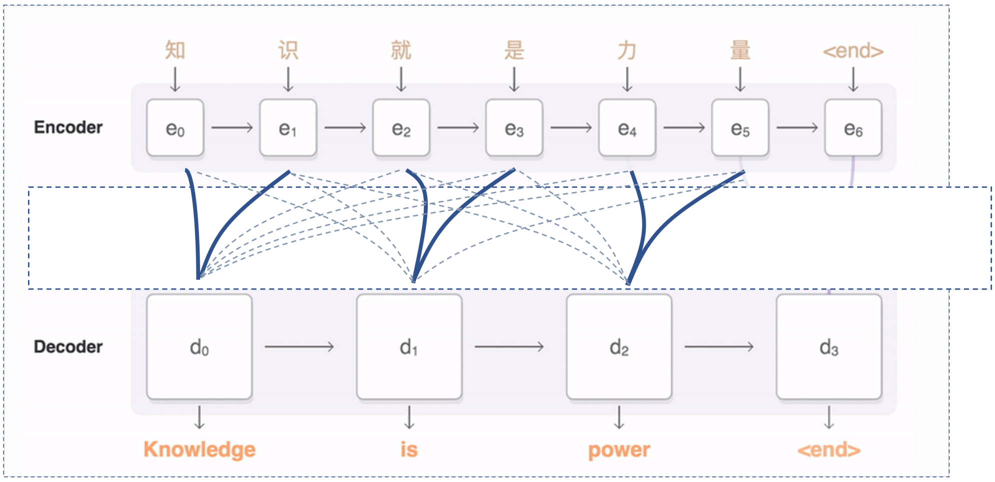


Fig. 2.3 An example showing how soft attention mechanism helps the model decoder focus on the relevant tokens in the input sequence.

former, illustrated in Figure 2.4, follows the encoder-decoder structure but relies entirely on the attention mechanism. Although the attention mechanism is exploited in the LSTM based models to help the network focus on the appropriate part of input when generating tokens, this process ignores the correlation within the sentence during encoding stage. The Transformer addresses this issue by applying a self-attention mechanism, which is the key innovation of it, which is computed as follows:

$$\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.2}$$

Here Q, K, V relates to the concept of query, key and value in the retrieval system. The general attention mechanism aims to find what elements (key) in the source input need to be paid attention to, given the target token (query). The relevance scores (value) between each key-value pair is usually computed through the dot-product operation. However in the self-attention mechanism, the query and key are from the same source (mostly the input sequence). As a result, the self-attention mechanism enables the system to accept input embedding from the previous encoder and weighs their relevance to each other to generate the encoding output.

This work also introduced the idea of Positional Encoding to make use of the order of the sequence, this is done by injecting relative or absolute position information into the word embedding. (We assume that readers have some knowledge about the Transformer architecture, we recommend (Alammar, 2020) as a detailed introduction of the Transformer) The Transformer architecture has gradually taken the place of RNN architecture in sequence modelling tasks for its outstanding performance, for example, the previously mentioned BERT model (Devlin et al., 2019) is one of the most famous applications of the Transformer architecture. Also, the Transformer architecture achieved leading performance on the WMT 2014 English-to-German translation task (Vaswani et al., 2017), and has since become the reference method in the machine translation field.

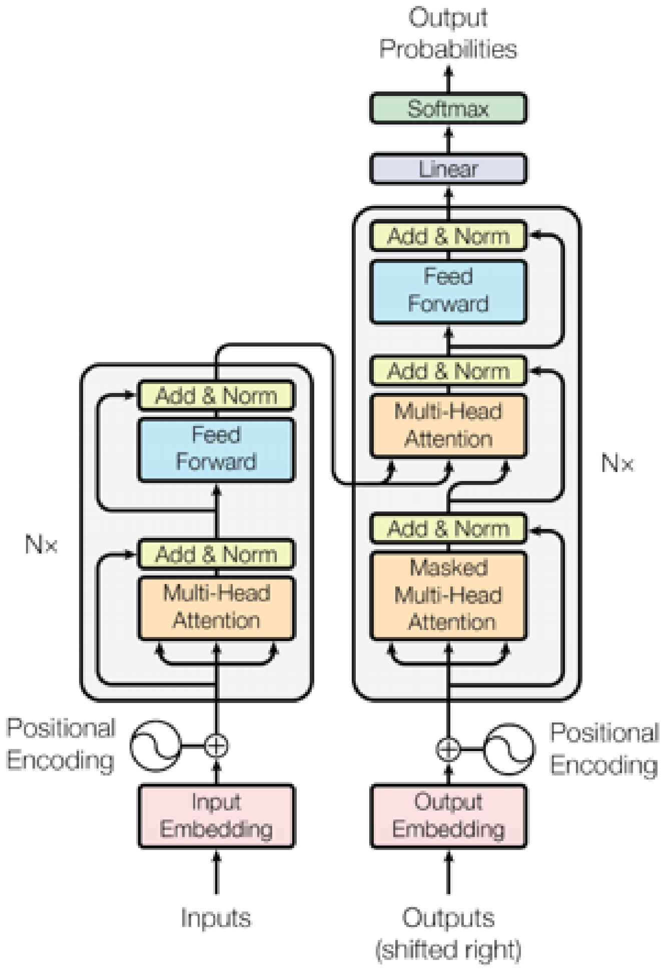


Fig. 2.4 Architecture of a standard Transformer network (Vaswani et al., 2017).

2.2 Video Analysis

One key feature of danmu comments is that they are presented conjointly with the video, this means that each comment will relate to a short clip of the video. According to our previous observation (see Figure 1.6), danmu comments are frequently associated with their closest video frames. This finding motivates us to utilize the information in the video content for our research question. In particular, we examine the methodologies in the field of video captioning, activity recognition and audio analysis.

2.2.1 Visual Representations

Similar to the text representation methods described in Section 2.1.1, various approaches have been proposed to represent human perceived visual signals.

The colour histogram is a statistical representation that can be regarded as an approximation of an underlying continuous distribution of colour values, this represents the number of pixels that

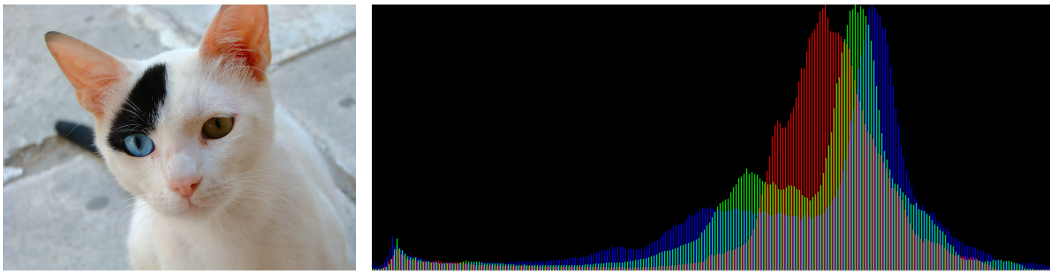


Fig. 2.5 Colour histogram (right) of the cat image (left) with x-axis being RGB and y-axis being the frequency.

have colours in each of a fixed list of colour ranges, that span the image's colour space, the set of all possible colours. The colour histogram can be built for any kind of colour space, although the term is more often used for three-dimensional spaces like RGB (red, green and blue, for a visual example see Figure 2.5) or HSV (hue, saturation and value). For monochromatic images, the term intensity histogram is usually used instead. The main drawback of colour histograms is that the representation ignores the spatial information and only focuses on the distribution of colours. Without spatial information, similar objects of different colour may be indistinguishable based solely on colour histogram comparisons, colour histograms can potentially be identical for two images with different object content which happens to share colour information.

In 1999, Lowe (1999) proposed scale-invariant feature transform (SIFT) to detect and describe local features in images. This method is implemented by first detecting key points in scale space, each point is then used to generate a feature vector that describes the local image region sampled relative to its scale-space coordinate frame. SIFT can robustly identify objects even among cluttered images, since the SIFT feature descriptor is invariant to uniform scaling, orientation, illumination changes, and partially invariant to affine distortion. The applications of SIFT include object recognition (Piccinini et al., 2012), robotic mapping and navigation (Hu et al., 2015), 3D modeling (Ohbuchi and Furuya, 2008) and gesture recognition (Sykora et al., 2014).

In the last decade, advances in deep neural models and large scale image classification datasets (*e.g.* Imagenet (Deng et al., 2009)) have heavily accelerated the development of visual object recognition methods, which has directly affected the performance of almost all computer vision tasks. Among these deep neural models, Convolutional Neural Networks (CNN) have proven to be effective in extracting visual features. This architecture proposed by LeCun et al. (1989) has achieved over 99% accuracy in the task of handwritten zip code digits recognition. Although it did not drawn much attention at that time, CNN has now become famous for its performance in computer vision tasks. Krizhevsky et al. (2012) proposed AlexNet, a CNN based architecture for image classification. This takes raw RGB values of images as input, the image feature in latent space is extracted by five CNN layers and another three fully connected layers are used to map the image feature into the classification probability scores. This method achieves an accuracy of 84.7% on the 2012 ImageNet LSVRC-2012 challenge as compared to the second-best with an accuracy of 73.8%. The success

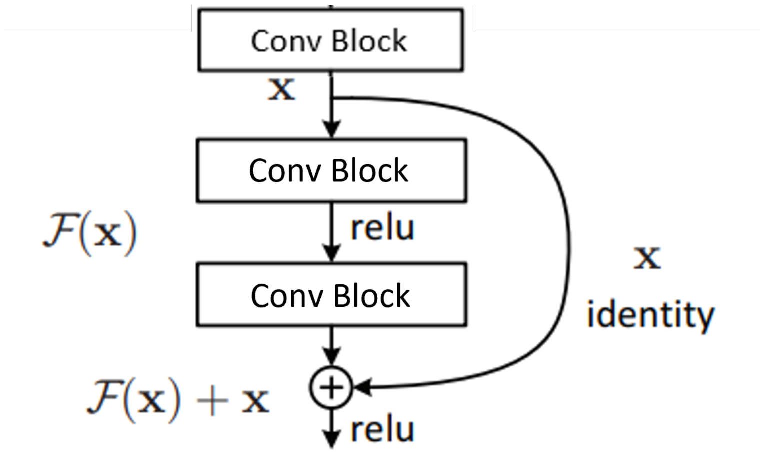

Conv Block
x
Conv Block
$\mathcal{F}(\mathbf{x})$
relu
Conv Block
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

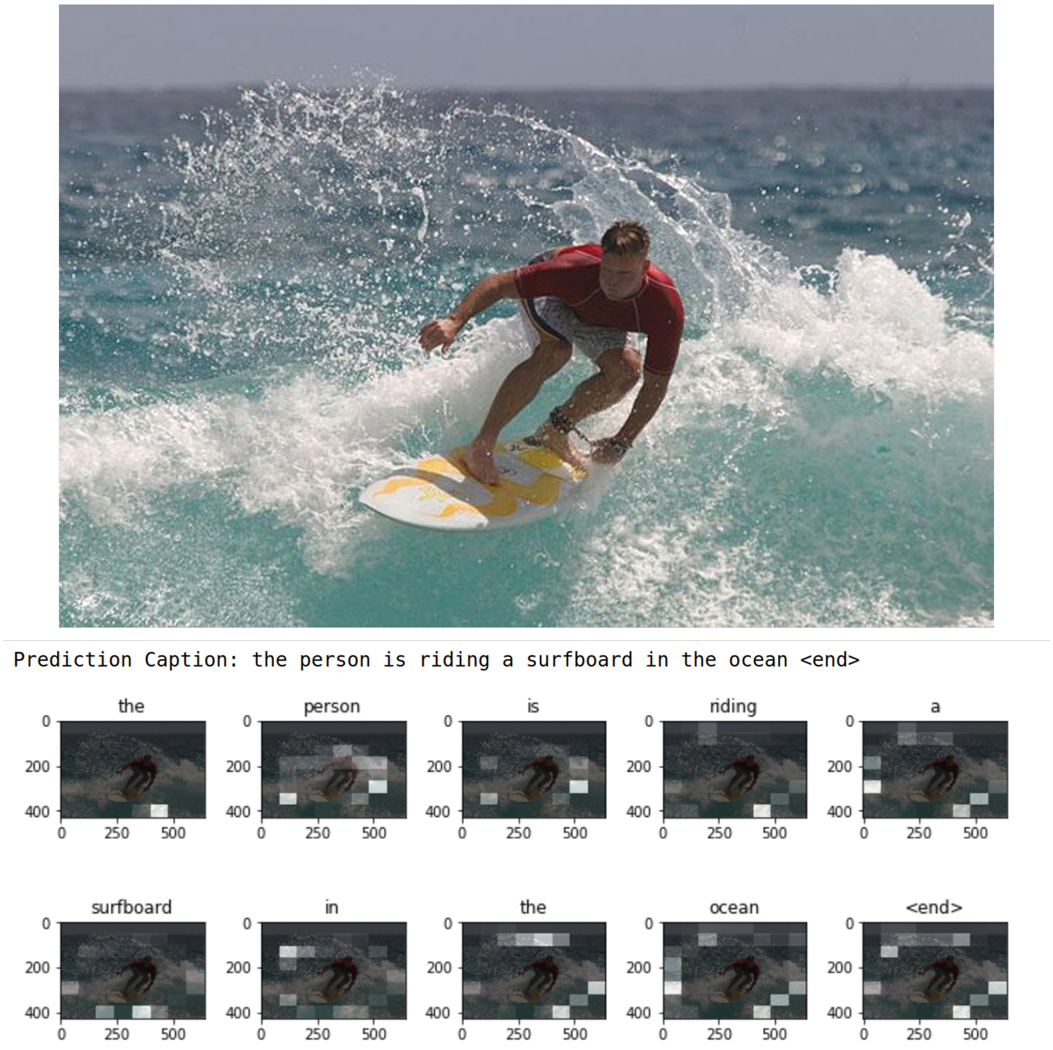
Fig. 2.6 Skip connection architecture in residual neural networks (He et al., 2016). The output from the first conv block x is added into the bottom conv block.

of AlexNet began the gradual dominance of CNN in the field of computer vision since then. The VGG model (Simonyan and Zisserman, 2014a) shows significant improvement by applying small kernels with fixed size throughout the model. ResNet (He et al., 2016) leverages the idea of skip connections (see Figure 2.6 for a visual illustration), which connects CNN blocks that are not linked together (usually skip one CNN block) by adding the output from the a previous layer into the current block. This approach achieved first place on the ILSVRC 2015 classification task. More recently, Dosovitskiy et al. (2020) proposed Vision Transformer (ViT), which applies a standard Transformer architecture directly to images. To do so, they split a two-dimensional image into patches and provide the one-dimensional sequence of linear embeddings of these patches as an input to a Transformer. When compared with CNN based architectures mentioned above, this purely self-attention based method achieved competitive performance on both mid-sized and large-scale image classification datasets.

These recent successes of deep learning models have greatly benefited all other computer vision tasks by providing high-quality visual representation. The output of the last pooling layer of a pre-trained ImageNet model is commonly assumed to be accurate and informative given the image input and has brought significant performance improvements to many computer vision tasks.

2.2.2 Video Captioning

An ongoing challenge in computer vision research is the semantic gap between low-level visual data and high-level abstract knowledge. To attempt to bridge the semantic gap, researchers start by generating text descriptions from static images. One good example to illustrate the process of image captioning can be found in (Vinyals et al., 2015), they address this task by using the output from the last layer of a pre-trained CNN model as the initial hidden state of a decoder RNN that generates the target sentence. An example of image captioning is presented in Figure 2.7.



```
Prediction Caption: the person is riding a surfboard in the ocean <end>
```

Fig. 2.7 An example of image captioning, with, on the bottom, the visualization of the decoder attention mechanism.

The task of video captioning resembles image captioning in its general goal (captioning natural language from visual signals) as well as model design. However, instead of processing a single image, a video captioning system takes a sequence of continuous video frames as input. Video captioning was already receiving intensive research attention even before the prevalence of deep learning. In 2002, Kojima et al. (2002) proposed to use hand-crafted features to detect visual concepts in a video and then generate a text description based on pre-defined templates. Such methods depend heavily on the templates and the generated sentences are always within fixed syntactical structures, not to mention that the design of hand-crafted features is also bounded for video understanding. The learning of video representations is the basis of video understanding, and in general involves both feature extraction and aggregation. The ultimate goal is to extract features from the video content, and then aggregate them both spatially and temporally to produce a compact video-level representation. Recent advances

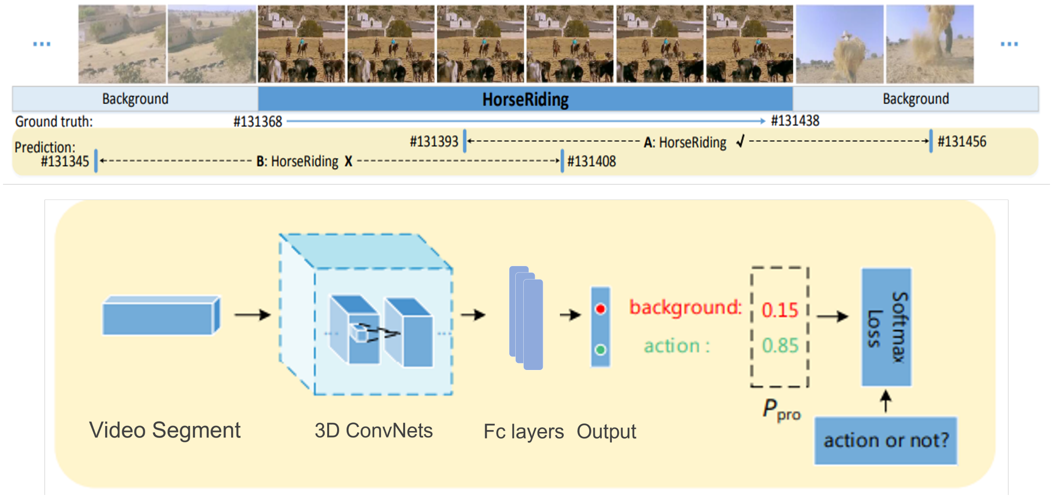


Fig. 2.8 An example of video activity recognition application from Shou et al. (2016), along with its model architecture.

in 2D and 3D CNNs have successfully improved state-of-the-art video representation learning from visual (He et al., 2016), audio (Hershey et al., 2017) and motion (Tran et al., 2015) information.

The decoder of sentence generation shares the same learning objectives and evaluation metrics with sequence generation tasks in NLG, such as text summarization and machine translation. Currently, there are several challenges in processing of video caption decoding. One of them is the problem of objective mismatch. The computation of sequence-level evaluation metrics such as BLEU Papineni et al. (2002) can not be directly optimized through back-propagation and gradient descent methods like normal objective functions. Rennie et al. (2017) attempted to solve this problem through Reinforcement Learning, they proposed a Self-Critical Sequence Training approach that can optimize any metric of interest through maximizing the expected reward of model samples and trains the model by using policy gradients. Considering that videos in real life are usually long and contain dense multi-modal information, how to leverage all the video content that is worthy of mention is also a challenging task. The work of Shen et al. (2017) was an early success in localizing video captions. This generates dense video captions that are spatially localized. With no spatial annotations, it first adopts multiple instance learning to extract visual concepts from video frames and then selects spatial region sequences which are then described individually by an LSTM-based decoder module.

2.2.3 Activity Recognition

Activity recognition aims at understanding and identifying activities that appear in the video, it has been a popular and active research topic for long time and contributes to a variety of high-level application areas such as human-computer interaction (Sharma and Verma, 2015), surveillance (Mabrouk and Zagrouba, 2018) and health monitoring (Nweke et al., 2019) systems. The case illustrated in Figure 2.8 is a good example of activity recognition, where the model tries to detect the action (horse riding) given the video segment. In this process, the model proposed in (Shou et al., 2016) first extracts a 3-dimensional video feature from a pre-trained CNN model, this video representation is then fed to a set of fully connected layers to generate the output probabilities of the action.

Most of activity recognition scenarios are recorded by surveillance cameras with a fixed angle, hence the viewpoint of the object and the background are fixed. Removing the redundancy of the background helps the model focus on foreground objects which are the key information in the activity recognition task. Cucchiara et al. (2003) applied a background subtraction approach to human activity recognition system for its accuracy and efficiency. The background image without any foreground object is first established. After that, the images in the analyzed video sequence are subtracted from the background image to obtain the foreground objects. The idea of background subtraction has been widely used by other researchers in the same field (Barnich and Van Droogenbroeck, 2010, Papazoglou and Ferrari, 2013). Activity recognition methods usually follow two key steps: object detection and activity mapping. The first step relies heavily on pre-trained image classification models (Simonyan and Zisserman, 2014b, He et al., 2016). The process of mapping a sequence of video frames to a certain type of action can also be improved from external knowledge. Such resources are built upon static visual signals and may be inadequate for activity recognition task. Since the recognition process involves the interpretation of a video scene containing a moving object, and relate the movements of the object to a certain type of action. Given such context, a major contribution is made by DeepMind, who released a large-scale, high-quality human activity recognition dataset, namely Kinetics (Carreira et al., 2019). One crucial post-processing step of action mapping is activity localization or activity alignment, which is about finding the most precise video clip that matches only to the target activity. A standard approach for activity localization is non-maximum suppression (Dai et al., 2017, Shou et al., 2016, 2017), which is also widely used in object detection to select the most appropriate bounding box for the object.

2.2.4 Audio Analysis

The audio signal in videos provides valuable additional information that is frequently leveraged in analysing video. In audio processing, mel-frequency cepstral coefficients (MFCC) (Logan, 2000) are often used as audio features. These are representations of the coefficients of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a non-linear mel scale of frequency. This audio representation is not very robust in the presence of additive noise, researchers try to normalise MFCC values in speech recognition systems to lessen the influence of noise. *E.g.* Tyagi and Wellekens (2005) proposes modifications to the basic MFCC algorithm to improve robustness, such as by raising the log-mel-amplitudes to a suitable power (around 2 or 3), which reduces the influence of low-energy components like environment noise.

The state-of-the-art in speech recognition is, once again, overwhelmingly dominated by recent advances in deep learning. In 2012, Hinton et al. (2012) demonstrated the success in using deep neural networks (CNN and RNN) for acoustic modelling in speech recognition. One year after this,

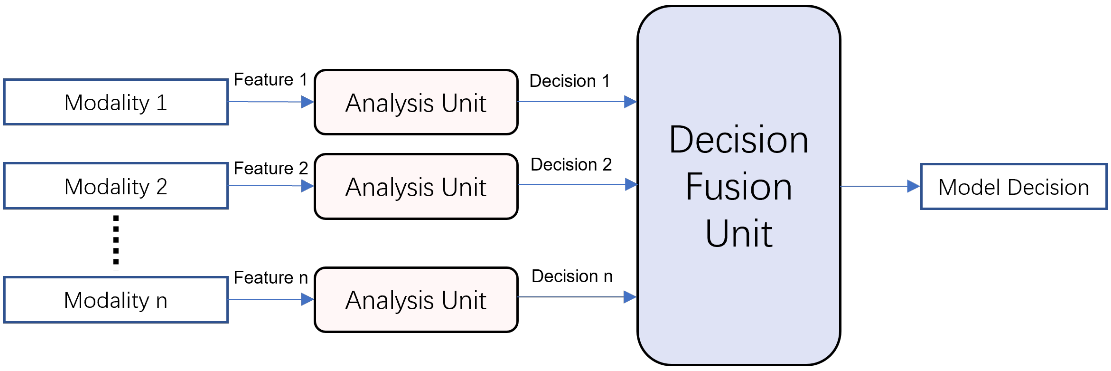


Fig. 2.9 The workflow of a decision-level multi-modal fusion process, the information from multiple modalities are fused in a late stage process.

Graves et al. (2013) showed a great improvement in recognition accuracy of the TIMIT phoneme recognition dataset by combining the multiple levels of representation (bidirectional RNN and LSTM). In order to deal with long-range temporal dependencies in the audio data, the Google DeepMind team proposed WaveNet (van den Oord et al., 2016), which is a CNN based generative audio model that takes raw audio waveform as input. This shows strong results on many tasks, including text-to-speech generation and speech recognition.

Note that the work in this thesis is not concerned with advancing the state-of-the-art in speech recognition, our objective is to use state-of-the-art methods to obtain audio transcripts from videos to support our creation of novel danmu comments.

2.3 Multi-Modal Fusion

In recent times, multi-modal fusion has gained much attention of many researchers due to the benefit it provides for various multimedia analysis tasks (Tsai et al., 2019, Rahman et al., 2019, Wang et al., 2016). The integration of multiple media, their associated features, or the intermediate decisions in order to perform an analysis task is referred to as multi-modal fusion. A multi-modal analysis task involves processing data from various modalities to obtain valuable insights, a complete understanding, or a higher-level perception of the data. The fusion of multiple modalities can provide complementary information and increase the accuracy of the overall decision-making process. For example, the fusion of audio and visual features have become more effective in many video analysis task (Iashin and Rahtu, 2020, Rahman et al., 2019). The fusion of different modalities is generally performed at two levels: decision-level (late fusion) and feature-level (early fusion):

2.3.1 Decision-Level Fusion

Late fusion or the decision-level fusion is the simplest and the most straightforward fusion method. It merges data after separate full processing in different uni-modal streams. As illustrated in Figure 2.9,

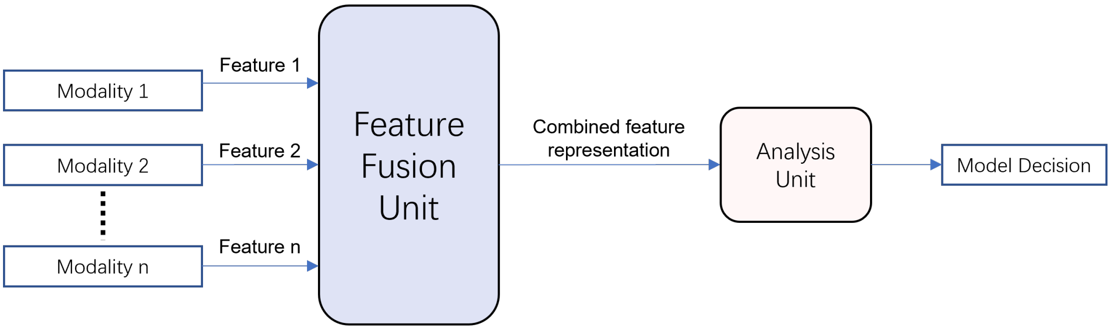


Fig. 2.10 The workflow of a feature-level multi-modal fusion process, the information from multiple modalities are fused in a early stage.

the analysis units first provide local decisions that are obtained based on individual features. The local decisions are then combined using a decision fusion unit to make a fused decision vector that is analyzed further to obtain a final decision about the task or the hypothesis. Several researchers have successfully adopted this decision-level fusion strategy. For example, Iyengar et al. (2003) obtained decisions from a face recognition module and a speech processing module, the uni-modal decisions were then fused through two approaches: the linear weighted sum and the linear weighted product. Walter et al. (2011) proposed a late fusion approach that leverages acoustic and bio-physiological data in detecting human emotional states. The data from different modalities were processed with corresponding modules like Hidden Markov model (HMM) and Multilayer Perceptrons (MLP) in their work. This work is an illustration of the advantage of the late fusion strategy: it enables us to use the most suitable methods for analyzing each single modality, such as HMM for audio data and CNN for image data. On the other hand, the drawback of the late fusion approach lies in its failure to utilize the feature-level correlation among modalities since each of the modalities is processed individually. This problem can be addressed by adopting feature-level fusion methods.

2.3.2 Feature-Level Fusion

In the feature-level or early fusion approach, the features extracted from input data are first combined and then sent as input to a single decoding analysis unit that performs the multi-modal analysis task (see Figure 2.10). For example, in the task of multi-modal machine translation, the feature fusion unit merges the multi-modal features such as source language context and visual signals into a fused feature vector which is taken as the input to the translation unit.

At the feature fusion stage, to exploit the cross-correlations between data streams, the fusion module needs to make sure that the data from different modalities is properly aligned. There are several choices for feature alignment: concatenation, direct mapping and self-attention: some fusion methods (Iashin and Rahtu, 2020, Huang et al., 2016) fuse the uni-modal representations from different modalities by concatenation, the features acquired from input data are adjusted to have similar dimensions and then concatenated to form a unified representation. This fusion approach is

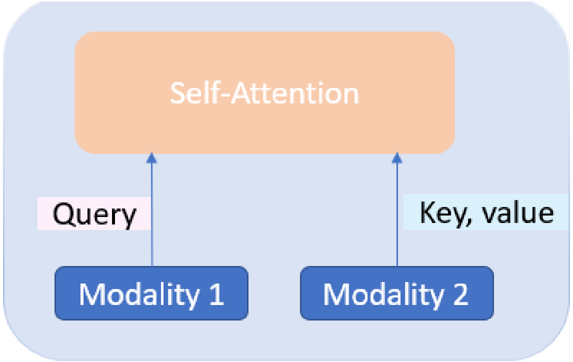


Fig. 2.11 The use of self-attention module in the fusion of two modalities.

intuitive and easy to implement and has been commonly accepted by many multi-modal analysis applications. (Libovický et al., 2016) use a fully connected layer to map the visual feature into textual feature space and then combine the features by a simple addition operation.

2.3.3 Feature-Level Fusion using Transformers

Despite originally being proposed for NLP tasks, Transformers have been recently proposed as multi-modal fusion models, due to their ability to model dense correlations among sequential signals.

The general idea is to use one modality as the "information seeker" (query) and the other one as "information provider" (key-value) in the self-attention block (see Figure 2.11), this enables the fusion module to fully explore the relations in-between modalities.

For the scenario that requires the fusion over two modalities, the fusion process can be implemented by sequentially attending new modalities into fused content (Ma et al., 2019). In such cases, pre-processed features are iteratively connected to each individual attention component. Alternatively, a variant has been proposed to explore cross-modal dependencies by fusing each possible combination of two modalities (Tan and Bansal, 2019, Tsai et al., 2019, Chaoqun et al., 2020).

The self-attention mechanism has shown to be effective in capturing the relevancy between different modalities (Tan and Bansal, 2019, Tsai et al., 2019, Chaoqun et al., 2020, Ma et al., 2019) with the transformer-based fusion approach now having become the state-of-the-art choice in multi-modal fusion.

2.3.4 Related Applications

Nowadays, huge amounts of multi-modal data with characteristics of high volume and great variety are becoming available. These data contain abundant inter-modality and cross-modality information, and have driven many research initiatives. Examples of multi-modal analysis tasks include multi-modal machine translation (Yao and Wan, 2020), visual question answering (Antol et al., 2015) and visual speech recognition (Neti et al., 2000).

Multi-modal machine translation (MMT) focuses on enhancing text-only translation with visual features, it has attracted considerable attention from both computer vision and natural language processing communities. MMT significantly extends the traditional text-based machine translation by integrating visual information as additional inputs (Specia et al., 2016, Caglayan et al., 2016, Yao and Wan, 2020). The assumption behind this task is that the translation is expected to be more accurate compared to purely text-based translation, since the visual context helps to solve data sparsity and ambiguity problems (Ive et al., 2019).

Visual question answering (VQA) has emerged as a prominent multi-discipline research problem in both academia and industry, and was established by the Microsoft team (Antol et al., 2015). The task is considered as the intersection of image captioning and text-based question answering. A VQA system takes as input an image and a free-form, open-ended, textual question about the image and produces a natural-language answer as the output (Yu et al., 2019, Lu et al., 2016), it resembles an MMT system in terms of the model architecture where they both have an emphasis on fusing textual and visual inputs. One difference is that a VQA system can be significantly improved by a knowledge-based approach, especially when the question requires extra knowledge beyond what is in the image provided (Yu et al., 2020, Zhu et al., 2020b). Both MMT and VQA share a similar fusion process with our research topic which involves consolidating textual and visual signals into natural language output. In contrast, one notable difference between automated danmu commenting and those two multi-modal analysis tasks is that the actual task of an MMT or VQA application is limited to a concrete scenario with a relatively small corpus, the scope of automated danmu commenting is not constrained with a specific scenario, thus the task corpus usually generalizes to a border range of domains.

Visual speech recognition (VSR) (Neti et al., 2000) also benefits from additional visual inputs, it aims to improve automatic acoustic-based speech recognition by exploring the visual modality. Unlike previously mentioned multi-modal applications where the added visual signals contribute to the task by providing semantically adequate supplementary information, in VSR the visual input is around the speaker's mouth region. The visual analysis involved in this task mainly focuses on extracting task-relevant information from lip reading. Existing work in VSR relies heavily on tracking facial motion (Galatas et al., 2012) or recognising lips poses (Afouras et al., 2018, Su et al., 2019, Mroueh et al., 2015) given the images of speakers. Apart from VSR, the concept of integrating visual and audio signals is also adopted in many video analysis tasks like video activity detection and video captioning. In automated danmu commenting, on top of the audio soundtrack extracted from the video, the audio representation can be supplemented by the text speech transcribed from the video.

2.4 Social Media Analysis

Research progress on analysing user-generated media on social platforms (*e.g.* Twitter, Weibo and YouTube) may provide practical examples that could potentially help the development of our research goal. In this section we examine literature on social sentiment analysis, social comments generation and highlight analysis of social videos.

2.4.1 Social Sentiment Analysis

The scale of social media platforms has grown very rapidly in recent years. As most of these platforms are reputation-driven, user opinions and ratings are increasing their importance in the evaluation of the services of the platforms. How to analyse user opinions effectively and efficiently has become one of the most popular topics in social media analysis.

One of the most widely examined areas of opinion analysis on social media is sentiment analysis. This uses NLP techniques to systematically analyse and identify emotional sentences. It is typically involved with the classification of the polarity of a given text. Existing approaches to sentiment analysis can be grouped into two main categories: knowledge-based methods and deep learning-based methods.

Knowledge-based Approaches

Knowledge-based methods classify text by the presence of unambiguous emotional words such as happy, sad, afraid, and bored. These methods rely heavily on external knowledge sources to extract emotions expressed in the text. This usually includes the use of sentiment lexicons, which contain pre-defined sentiment polarity information of frequent words, then the sentiment of the sentence or documents to be classified is calculated by aggregating the sentiment scores of each individual word. The most commonly used sentiment lexicons are WordNet (Miller, 1995) and SentiWordNet (Sebastiani and Esuli, 2006). In these sentiment lexicons, a word is usually assigned with expert-labelled sentiment orientation scores that show the degree of positive, negative and neutral of the word. Similar efforts have been made in creating sentiment dictionaries. In very early work, Stone et al. (1966) built the general inquirer (GI), which provides comprehensive details of each word including sentiment polarity, lexical category and antonym. Pennebaker et al. (2001) released an open tool for text analysing called Linguistic Inquiry and Word Count (LIWC), this produces a sentiment score by using a number of regular expressions to describe different categories of sentiment word patterns. Taboada et al. (2011) propose the Semantic Orientation CALculator (SO-CAL) which uses sentiment dictionaries in sentiment classification, they consider the intensification and negation of the words. The performance of SO-CAL is consistent even with unseen data. (Bhuiyan et al., 2017) applies a lexicon-based sentiment analysis method on YouTube comments. They further use the analyzed user sentiment result to assist a video retrieval system in finding the most relevant and popular video. Most of the knowledge-based sentiment classification methods require high-quality sentiment lexicons, this limits the performance of these methods as the coverage of a pre-defined lexicon is usually limited. In such cases, knowledge-based approaches can not easily generalise to new unseen domains. Researchers have started to use machine learning and deep learning techniques to improve the weaknesses of the existing knowledge-based methods.

Learning-based Approaches

Instead of looking at the sentiment scores at a word level, most learning-based methods associate the entire sentence or the document with a sentiment label. In this process, the text source is usually transformed into a text representation, such as tf-idf scores or word embedding, and then converted to the likelihood scores of sentiment orientation using a Support Vector Machine (SVM) or deep neural networks. (Zhang et al., 2015) use Word2Vec to extract semantic features of Chinese text sources and predict sentiment scores with an SVM module, they report over 90% accuracy in their study. Since the advent of deep neural networks, new studies in the field are emerging. Ramadhani and Goo (2017) propose deep learning systems for sentiment analysis of tweets. Cunha et al. (2019) apply deep neural networks on YouTube comments, they use a pair of CNN layers to extract text features and a fully connected layer to classify video comments between positive, negative or neutral.

Multi-modal Sentiment Analysis

Text-based sentiment analysis is currently widely used for user satisfaction assessment and brand perception analysis. The arrival of multi-modal data streams has motivated researchers to start exploring multi-modal sentiment analysis. The basic assumption is that sentiment can be detected through multiple modalities, such as facial expressions and vocal displays, these signals can supplement transcripts or textual content in detecting user sentiment. The work of Pérez-Rosas et al. (2013) is a practical example of leveraging multi-modal signals in sentiment detection. They built a dataset by collecting videos about product reviews on YouTube channels. Then the video speech, video frames and the soundtrack were extracted as multi-modal inputs to detect the sentiment of the reviewer. In particular, they processed the facial expressions from the video frames as visual features. Experiment with their dataset achieved a 10% error reduction with the use of multiple modalities when compared with pure text-based methods. Zadeh et al. (2017) proposed to solve multimodal sentiment classification on YouTube movie reviews videos, their method learns both the intra-modality and inter-modality connections in an end-to-end neural network.

In summary, social sentiment analysis is not closely related to our current research topic, however, the multi-modal approaches in this field provide good insights into the analysis of the user opinions on social video platforms.

2.4.2 Social Comments Generation

Automated creation of human-like comments is one of the most challenging tasks in social media analysis, this requires the ability to both understand social media content and communicate via natural language. In the previous years, the majority of existing literature in this field follows a text-based dialogue structure, the social media response to be generated is typically associated with existing user comments. Ritter et al. (2011) proposed a data-driven approach to generate comments on Twitter. They aimed to generate comments given previous human tweets. This task is purely text-based and very close to machine translation in its form, this observation motivated them to utilize machine translation techniques in comment generation. They adopted a SMT (statistical machine translation) model and used standard machine translation metrics like BLEU for evaluation. In this work, the problem of lexical repetition (which refers to the SMT system that tends to repeat the words from the input text) is carefully investigated, to alleviate this phenomenon they proposed a novel feature to penalize lexical similarity which is measured by Jaccard distance. Also, to test the effectiveness of machine translation metrics (BLEU) in evaluating social media commenting, they measured the correlation of BLEU score and the manual judgments and showed that the BLEU score has a weak positive correlation with annotator judgments.

Deep learning-based methods have recently been widely leveraged in social media commenting (see Section 2.1.2). These methods are effective in aligning words between input and output and can better capture sequential information from natural language. Most deep learning NLG systems learn from corpora by minimizing entropy loss between the output and a groundtruth, this prompts neural models to generate trivial responses like “me too” and “I don’t know” with high frequencies in conversations. To prevent the commenting system from producing this type of tedious content, various approaches have been proposed. Xing et al. (2017) used an LSTM based sequence-to-sequence model to generate natural responses in Baidu-Tieba (the largest Chinese forum-based communication platform) with the help of external knowledge. They trained a separate LDA topic model (Blei et al., 2003) that offers context-relevant topic words. This topic-related knowledge guided the commenting system to form informative and interesting responses. For evaluation metrics, instead of using machine translation metrics, they used Perplexity (measuring how confident the model is in predicting a response) and Distinct-n (number of distinct n-grams in the generated responses) to evaluate the commenting system to give better measurements of diversity and relevancy of conversational sentences.

Some researchers have attempted to guide the NLG model in producing non-trivial content by applying reinforcement learning strategies. In the work of Li et al. (2016), reinforcement strategies are applied to reward sentences that display three expected conversational properties: informativeness, coherence, and ease of answering. These properties are digitized individually and are used as rewards during the optimization of an LSTM based encoder-decoder network. This method is able to capture global properties of the given conversation context and generate non-trivial utterances.

Transformer architectures are showing increasingly promising results in many NLG tasks mentioned earlier. Likewise, the Transformer architecture is frequently applied in the task of video comment generation. Wang et al. (2020c) use a Transformer to encode video frames together with the previous dialogue in order to generate dialogue related target comments. They use pre-trained BERT weights to initialize the text embedding and achieved state-of-the-art performance in the Visual Dialog v0.9 and v1.0 datasets (Das et al., 2017). Jin et al. (2019) addresses automated commenting in dialogues around video clips collected from YouTube by using a cross transformer. The cross transformer module learns fine-grained and comprehensive interactions both inside and between the input modalities (textual dialogue and visual video frames). In this work, they introduce a novel progressive

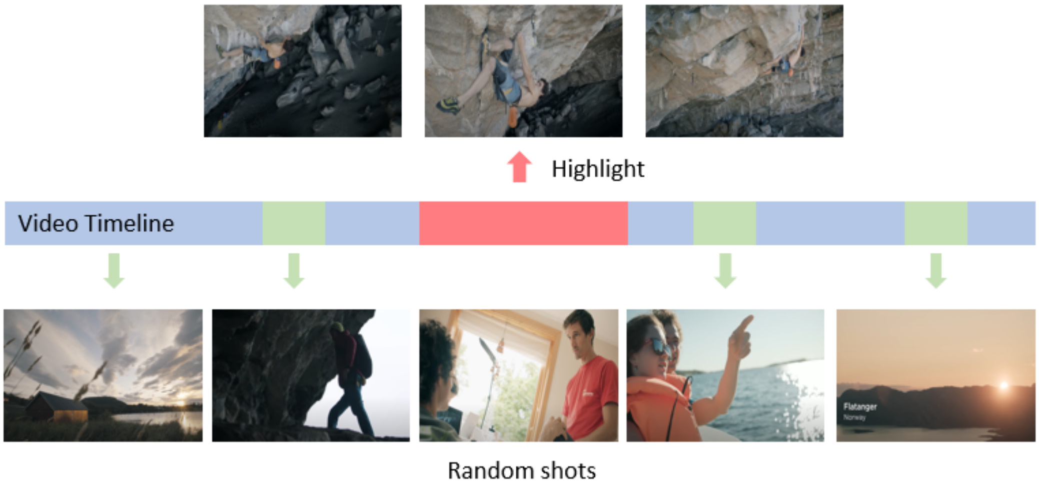


Fig. 2.12 Video frames from a rock-climbing video, the highlighted frames of the actual rock-climbing content are listed on the top, other non-highlight frames that are randomly selected from the video are presented below.

inference mechanism for encoding multi-modal inputs. The inference module progressively updates the encoder's state based on dialogue history and video content until the information is sufficient and unambiguous.

In this section, we present state-of-the-art research in social comment generation. Many methods including rule-based systems, SMT, topic models and deep learning approaches have been proposed to solve this task. One trend we can observe is that the current work in this field often requires the analysis of multi-modal social media content compared against the text-based dialog system. Among the approaches introduced, transformer-based methods have recently become the most popular choice for their good performance in multi-modal fusion.

2.4.3 Social Videos Highlight Analysis

In the last chapter, we raised a research question ([RQ2]: Can we automatically identify appropriate locations in video timelines to insert comments?). One solution to this research question is finding the highlights of the video as the preferred locations for adding comments, we thus look at related literature on social video highlight analysis.

Social video sharing platforms are becoming increasingly prevalent, indexing, organizing, and even browsing such huge traffic consumes enormous resources. To mitigate this overload, video highlight detection has been proposed to analyse social videos with a focus on capturing the user's attention. The general goal of the task is to extract the most interesting and important clips of a video. Figure 2.12 provides an example of extracting highlight moments from a rock climbing video.

A well-extracted highlight can cover the most important moments of the video and thus enhance the viewing experience for users. This task requires a deep understanding of the video content that goes beyond the semantic level, since the definition of highlight could be very subjective across different domains. When compared to the task of video activity recognition, where the target event is a pre-defined action with a clear description, the highlights that need to be analysed or detected are usually not clearly defined or described concretely, this ambiguity makes the task ever challenging. Generating highlight clips is also different from the task of video summarization which tries to select a diverse and representative summary of a video rather than a brief interesting moment.

Existing approaches to highlight detection can be categorized into two broad strategies based on the source of training signals: supervised and semi-supervised.

2.4.4 Supervised Highlight Detection

This strategy considers highlight detection as a supervised learning task. The video dataset used is manually annotated where each segment of a video is labelled as a highlight or not. Supervised approaches are commonly adopted in detecting highlight of domain-specific sports videos, like soccer (Assfalg et al., 2002), baseball (Xiong et al., 2005) or cricket (Tang et al., 2011). In these methods, video segments are first analyzed and processed into visual features, and then mapped to the target highlight type. In (Tang et al., 2011), low-level visual features like colour histogram or histogram of oriented gradients (HOG) are extracted as video representation, then a linear support vector machine (LSVM) classifier is trained to classify input video features into one of two categories: highlight clips or non-highlight clips. In a more recent case (Godi et al., 2017), they proposed to use deep neural nets in detecting highlights from ice-hockey videos. In this work, pre-trained CNN visual features were used to represent video clips with fully connected layers as the classifier.

While the methods trained on well-annotated supervised signals have the advantage of good discriminative power, these models usually suffer from heavy, non-scalable supervision requirements which means that they are unable to adapt to other domains.

2.4.5 Semi-Supervised Highlight Detection

The second strategy instead regards highlight detection as a weakly supervised learning task. Given the videos, the system discovers what appears commonly among the training samples, and learns to detect highlights indicators as training supervision.

Most of the work that focuses on user-generated web videos adopts this strategy. Xiong et al. (2019) propose to use video duration as a weak training signal for detecting highlights of YouTube and Instagram videos. The authors observe that individual clips from a short video are likely to be highlights of that video, whereas clips from long videos are unlikely to be identified as highlighted scenes. This observation is used to obtain groundtruth labels when comparing the highlight probability between two random clips. Yang et al. (2015) propose an unsupervised approach for generating highlight clips from edited web videos. The key idea is that the most significant sub-events within a video are commonly present among edited videos while less interesting ones appear less frequently. The system learns what is a highlight based on commonalities among videos in the dataset, in this process, the video clips are represented using pre-trained 3D-CNN features, then a discriminative model for highlight detection is built by an LSTM based auto-encoder.

Returning to our research focus, danmu videos have a unique advantage over other types of social videos, the distribution of danmu comments over the video timeline potentially reflects user attention and can be used as the signal of highlights. This free, rich and precise labelling has promoted danmu video highlight analysis, several methods have been proposed to analyse the highlights of the danmu videos. One notable contribution is that of Zheng et al. (2020), they propose to predict highlights for newly published videos by taking the video frames and the existing comments into two separate LSTM modules and using them to jointly predict the number of comments published in a video segment.

2.5 Conclusions

In this chapter, we reported an extensive literature review of research topics including natural language representation and generation, multi-media signal fusion and social content analysis relevant to our research.

During the investigation of the area of NLG and multi-media processing, we observed that the Transformer architecture is taking over from other neural architectures (LSTM and CNN) and is frequently considered as a universal solution to tasks spanning a wide range of fields, such as text processing, text generation, image processing and even multi-modal fusion. This insight is valuable in terms of selecting neural architectures for our later explorations towards our research questions. We also note that multimedia analysis is a fast-evolving field and the most recent advances that can potentially contribute to our research objective may not appear in this Chapter. Another key finding that may help us in solving the second research question (RQ2: Can we automatically identify appropriate locations in video timelines to insert comments?) arises from our analysis of the relevant work in social media highlight analysis, where the general goal is to detect the most interesting and important clips of a video. Coincidentally, this task provided us with a new direction in automated danmu commenting, as in danmu videos, the highlights could potentially serve as preferred locations for automated commenting. In the next chapter, we introduce the detailed functionalities of danmu videos and examine the available data streams on danmu. Following that we present our danmu video collection which expands a public available dataset (Ma et al., 2019).

Chapter 3

Danmu Comments Dataset

Danmu comments or "danmu" have become one of the key features underpinning the recent success of video-sharing platforms in Asia. Media consumption in platforms using danmu type comments is highly interactive, with multi-modal data streams which can become richly labelled, and opens the path to multiple new research strands for online video content, including automated highlighting (Zheng et al., 2020), recommendation (Xu et al., 2017a) and conversational engagement (Ma et al., 2019).

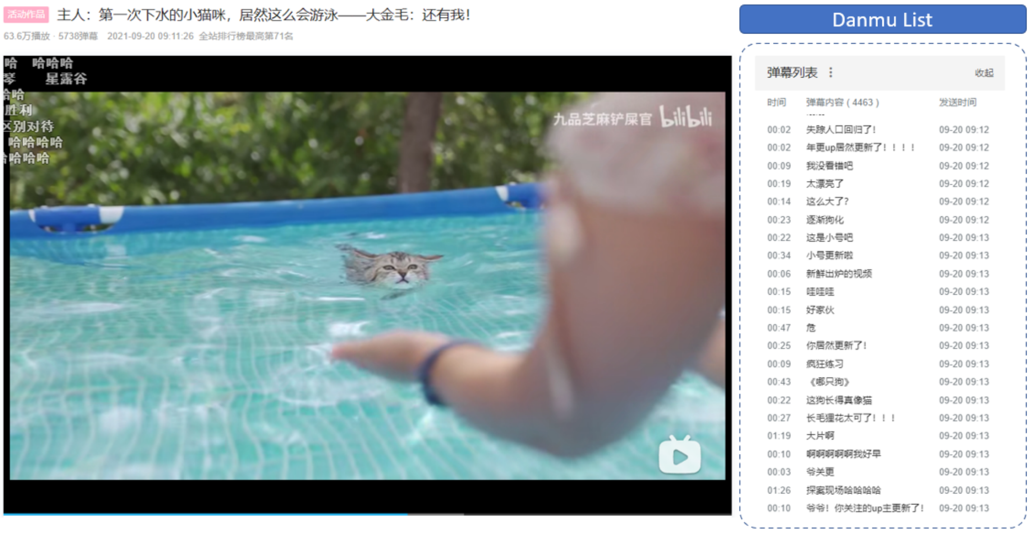
In this chapter, we first outline the structure of the data stream available for danmu videos on danmu video platforms. Then we present an overview of one public danmu video dataset proposed by Ma et al. (2019), and finally we introduce our expanded version of this danmu video dataset which is used in our following research investigations.

3.1 Anatomy of Danmu Comments on Bilibili

Most of the existing danmu video datasets (Lv et al., 2019a, Ma et al., 2019, Wang et al., 2020b) are sourced from Bilibili, which is one of the most popular danmu video platform in China. For a danmu video in Bilibili, the raw video stream, the danmu comments list and even video meta-data can be downloaded using a web crawler. An example of the danmu video interface along with its danmu list is shown in Figure 3.1. As we can see, comments are embedded in the video playback screen. Note that their order of appearance is based on their associated video display timestamp rather than their actual publication date.

For each of the comments that appear in the video, we have access to the video display timestamp tag, indicating where it should appear in the video, and the publication date, indicating when the comment was posted to the platform. Some display attributes of danmu comments (*e.g.* location, font, or the floating speed) are computed in real-time. The information about the sender of the danmu comment is not made explicitly available on the website.

To better illustrate this mechanism, Figure 3.2 shows six comments selected that appear in the same video frame and are sorted by their publication date. Even though some of these comments have



活动作品 主人：第一次下水的小猫咪，居然这么会游泳——大金毛：还有我！

63.6万播放 · 5738弹幕 2021-09-20 09:11:26 全站排行榜最高第71名

时间	弹幕内容 (4463)	发送时间
00:02	失踪人口回归了！	09-20 09:12
00:02	年更up居然更新了！！！！	09-20 09:12
00:09	我没看错吧	09-20 09:12
00:19	太漂亮了	09-20 09:12
00:14	这么大了?	09-20 09:12
00:23	逐渐狗化	09-20 09:12
00:22	这是小号吧	09-20 09:13
00:34	小号更新啦	09-20 09:13
00:06	新鲜出炉的视频	09-20 09:13
00:15	哇哇哇	09-20 09:13
00:15	好家伙	09-20 09:13
00:47	危	09-20 09:13
00:25	你居然更新了！	09-20 09:13
00:09	疯狂练习	09-20 09:13
00:43	《哪只狗》	09-20 09:13
00:22	这狗长得真像猫	09-20 09:13
00:27	长毛狸花太可了！！！	09-20 09:13
01:19	大片啊	09-20 09:13
00:10	啊啊啊啊啊我好早	09-20 09:13
00:03	爷关更	09-20 09:13
01:26	探案现场哈哈哈哈	09-20 09:13
00:10	爷爷！你关注的up主更新了！	09-20 09:13

Fig. 3.1 The original Bilibili danmu video interface with the video presented on the left and danmu comment list on the right. Each danmu comment is also labelled with its video display timestamp (on the left of the comment) and publishing dates (on the right of the comment).

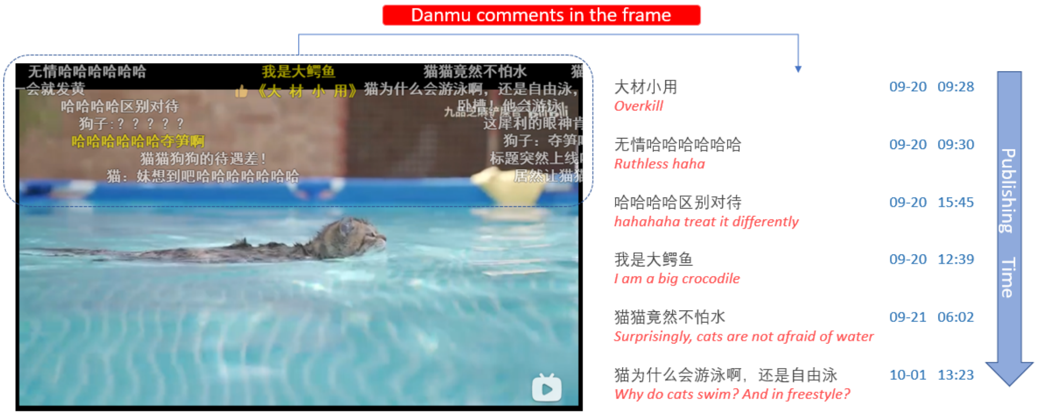
been posted a few days apart, they still appear together in the video frame as their display timestamps are close to the current playback timestamp (07:19).

Apart from danmu comments, meta-data of danmu videos are also presented on the website and could potentially be leveraged in danmu video analysis. As presented in Figure 3.3, video meta-data includes video statistics, titles and video tags. Video statistics like the number of views, likes and forwards are considered to be critical measurements of video reputations. These statistics have become precious resources that can be exploited by the platform and the research community (Lv et al., 2019b) to analyse these social videos. Video title and tags are textual descriptions of the video. These messages offer highly accurate and representative labels for the video data and can be utilized directly in many danmu related research tasks like sentiment analysis or danmu generation.

A number of the danmu collections have been created for use in previous studies on danmu video commenting. However, most of these are not publicly available (e.g. Lv et al. (2019b), Wang et al. (2020b).). The only publicly available dataset at the start of our project was the Livebot dataset Ma et al. (2019), which we used as a starting point for our research.

3.2 LiveBot Dataset

The Livebot dataset was released at the end of 2018 for the automated danmu commenting task. This is the first large-scaled danmu video dataset on the Internet and contains more than two thousand danmu videos. The publication of the Livebot dataset stimulated a series of follow-up studies including (Zhang et al., 2020, Chaoqun et al., 2020) on the task of automated danmu commenting.



大材小用
Overkill 09-20 09:28

无情哈哈哈哈哈哈
Ruthless haha 09-20 09:30

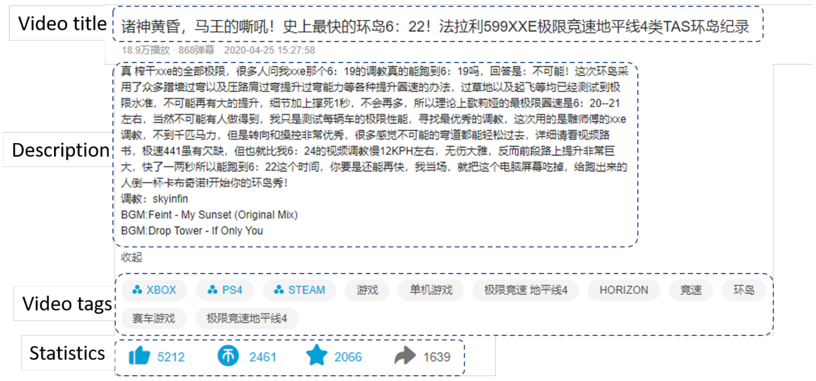
哈哈哈哈区别对待
hahahaha treat it differently 09-20 15:45

我是大鳄鱼
I am a big crocodile 09-20 12:39

猫猫竟然不怕水
Surprisingly, cats are not afraid of water 09-21 06:02

猫为什么会游泳啊，还是自由泳
Why do cats swim? And in freestyle? 10-01 13:23

Fig. 3.2 The comments in the video frame (left side) are translated and sorted by the publication date (right side).



诸神黄昏，马王的嘶吼！史上最快的环岛6：22！法拉利599XXE极限竞速地平线4类TAS环岛纪录

18.9万播放 · 868弹幕 2020-04-25 15:27:58

真 榨干xxe的全部极限，很多人问我xxe那个6：19的调教真的能跑到6：19吗，回答是：不可能！这次环岛采用了众多蹭墙过弯以及压路肩过弯提升过弯能力等各种提升圈速的办法，过草地以及起飞等均已经测试到极限水准，不可能再有大的提升，细节加上撑死1秒，不会再多，所以理论上歌莉娅的最极限圈速是6：20--21左右，当然不可能有人做得到，我只是测试每辆车的极限性能，寻找最优秀的调教，这次用的是雕师傅的xxe调教，不到千匹马力，但是转向和操控非常优秀，很多感觉不可能的弯道都能轻松过去，详细请看视频路书，极速441虽有欠缺，但也就比我6：24的视频调教慢12KPH左右，无伤大雅，反而前段路上提升非常巨大，快了一两秒所以能跑到6：22这个时间，你要是还能再快，我当场，就把这个电脑屏幕吃掉，给跑出来的人倒一杯卡布奇诺!开始你的环岛秀！
调教：skyinfin
BGM:Feint - My Sunset (Original Mix)
BGM:Drop Tower - If Only You

收起

XBOX PS4 STEAM 游戏 单机游戏 极限竞速 地平线4 HORIZON 竞速 环岛 赛车游戏 极限竞速地平线4

5212 2461 2066 1639

Fig. 3.3 Danmu video meta-data. including title, description by the video uploader, video tags and video statistics.

3.2.1 LiveBot Dataset Formation

In order to collect representative and high-quality danmu videos from Bilibili, the most commonly searched keywords on the platform were used as queries to the platform search engine, and the top 10 pages of the video search results were then crawled. The queries covered 19 categories, including pets, sports, animation, food, entertainment, technology and more. After the removal of short videos and less-commented videos, a total of 2,361 videos remained to form the dataset. For each video, the corresponding danmu comment list and the video title were collected, the danmu comments that appeared in the videos were also collected along with their video display timestamp and publication date. For this dataset, audio soundtracks of danmu videos were not specifically extracted or processed. Videos were then split into training, development and test sets. A detailed overview of the dataset statistics is shown in Table 3.1.

To benefit future research, both the raw danmu video dataset and the processed dataset were released by (Ma et al., 2019). For the processed dataset, all 895,929 comments were tokenized using

Table 3.1 Training, development and test sets statistics of the original Livebot dataset, showing the number of videos and danmu comments, the average number of words per comment, average durations of the videos and the average number of comments per second.

Statistics	Training	Dev	Test	Total
#Videos	2,161	100	100	2361
#Danmu	818,905	42,405	34,609	895,929
#Danmu / Video	378.94	424.05	346.09	379.47
Avg.Words/Danmu	5.39	5.85	5.58	5.42
Avg.Duration (mins)	2.88	3.01	3.00	2.89
Avg. #Danmu/s	2.19	2.34	1.92	2.18

Table 3.2 The Livebot dataset has one key issue: several comments seem to appear in both the training and test set.

Comments	Translation
像我这么瘦的可能效果不会太明显，各种无器械动作交杂着做两个多月才有了明显的变化，还不是很大	It might not be obvious for skinny people like me, there are only minor changes after two months of exercise.
这样看不出，第一和最后一天再对比下，长肌肉不是做了多少，是休息恢复，增长多少，不建议天天练	Can not tell anything from this, muscle growth is about resting and recovering rather than work out everyday.
100个考验耐力，到后期，强度就不高了，复合俯卧撑最好	Doing 100 requires endurance not strength, compound push-up is the best.
每天100个俯卧撑100个仰卧起坐跑步10公里坚持3年然后再把头发剃光滑稽	100 push-ups 100 sit-ups 100 squats and a 10km run every single day for three years then shave your hair lol.
练肌肉最费钱，想练快就每天吃低脂牛肉，配合锻炼，半年就有显著变化	Muscle gain is expensive, regular exercise with low-fat beef and you will see the changes in half a year

Jieba [1](a popular Chinese word segmentation tool), for each of the videos, one video frame per second was extracted and ResNet-18 (provided by the PyTorch package, pre-trained on full ImageNet dataset) was used to extract visual features.

3.2.2 Issues with the Livebot Dataset

We carefully examined the released Livebot dataset, specifically, we checked the overlapped comments across the training and test set of the given processed dataset. We found that 5,436 out of 17,771 comments in the test set that also appear in the training set. Although some popular comments can be expected to appear in multiple videos, after manually checking the provided dataset we found that there are a number of identical videos assigned with different video ids that appear in both the training and test sets. Table 3.2 lists several examples we found of this situation. In the raw dataset, we use video titles to uniquely identify a video and found that 38 video titles appear more than once in the raw dataset. After manual inspection of these duplicates, we found both the video content and the danmu comments are also identical. These overlaps with the dataset will cause data leakage during the evaluation process since some of the data in the test set has already been seen during training.

[1] https://github.com/fxsjy/jieba

Statistic	Train	Dev	Test	Total
#Videos	2,122	100	100	2,322
#Danmu	788,645	34,581	34,767	857,993
#Danmu / Video	371.6	345.81	347.67	369.47
Avg. Words /Danmu	5.37	5.45	5.36	5.38
Avg.Duration (mins)	2.88	3.11	2.95	2.89
Avg. #Danmu/s	2.15 2.19	1.85	1.96	2.13

Table 3.3 Statistics of the updated Livebot dataset.

We addressed the above issue by removing the duplicate videos in the raw danmu video dataset, full details are contained in (Wu et al., 2020b). Due to the lack of video mapping between the raw dataset and the processed dataset, we modify the raw dataset and split these videos into training, development and test set by following (Ma et al., 2019). After the correction, our dataset consisted of 2322 unique videos, the statistics of our new Livebot dataset are summarised in Table 3.3.

3.3 Proposed Extended LiveBot Dataset

3.3.1 Dataset Collection

In order to support our later work on the topic of automated danmu commenting, as well as to explore the influence of dataset size over the behaviour of the commenting system, we constructed a larger dataset with 4,672 videos and 2,789,360 danmu comments, which is publicly available [2]. Part of the data (2,322 videos and 857,993 comments) comes from the corrected Livebot dataset *Response to Livebot* dataset (Wu et al., 2020b). We added another 2,350 videos from the same source (Bilibili) to the dataset. Considering the Livebot dataset is mainly themed around daily activities (*e.g.* pets, piano and basketball), to keep it consistent, our additional videos were selected by having a web crawler pick the 100 most popular "Daily Life" category videos of the recent three days every day for two months, this enables us to cover more topics within the category of "Daily Life". The leader-board of the most popular videos is presented in Figure 3.4, on average each video has roughly 2.3 million views and 8,400 danmu comments. The popularity of a video is measured based on recent user feedback including the number of danmu comments, views etc. The statistics of our collected dataset are shown in Figure 3.4

In comparison, our collection has generally notable advantages over the Livebot dataset in two aspects: topic range and comment intensity. The videos in the Livebot dataset are retrieved based on popular keywords and only contain specific topics, *e.g.* piano or pets. Regarding the span of the topics, our collection covers a broader range by including the popular videos in the "Daily Life" category. This is the largest category on the platform and includes most of the topics that appear in the Livebot dataset plus plenty of other interesting topics such as TV shows, product reviews, cooking, knowledge

[2] github.com/fireflyHunter/Cold-Video-Danmu-Generation

Statistics	Training	Dev	Test	Total
#Videos	2150	100	100	2350
#Danmu	1,760,695	89,065	81,607	1,931,367
#Danmu/video	818.92	890.65	816.07	821.85
Avg. Words /Danmu	5.19	5.35	5.38	5.21
Avg. Duration (mins)	4.33	4.38	4.19	4.33
Avg. #Danmu/s	3.15	3.39	3.24	3.17

Table 3.4 Training, development and test sets statistics of our collected videos.

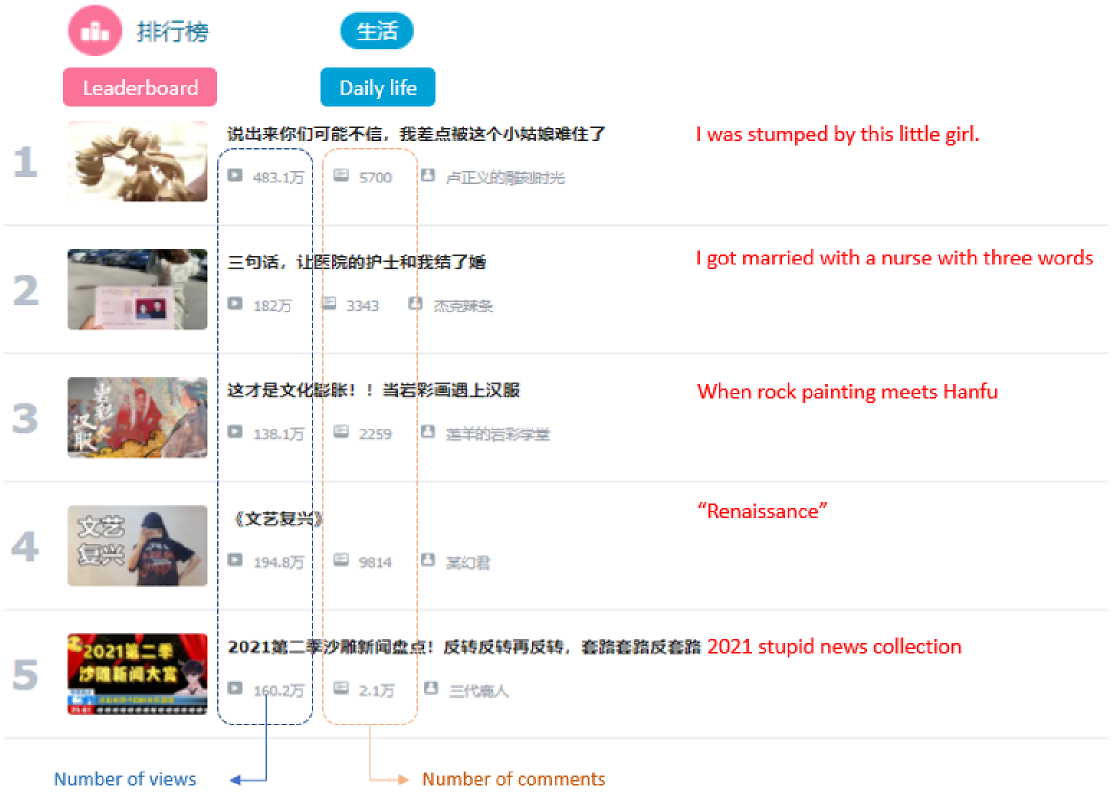
Statistics	Training	Dev	Test	Total
#Videos	4,272	200	200	4672
#Danmu	2,549,340	123,646	116,374	2,789,360
#Danmu/video	596.75	618.23	581.87	597.03
Avg. Words /Danmu	5.26	5.39	5.37	5.27
Avg. Duration (mins)	3.61	3.7	3.6	3.61
Avg. #Danmu/s	2.75	2.78	2.69	2.75

Table 3.5 Training, development and test sets statistics of the complete dataset.

sharing, handmade arts etc. The diversity of our collection is shown in Figure 3.5. On the other hand, in our data crawling process, we collected the most popular videos over two months. This allows us to only include intensively commented videos in our collection. The videos in the Livebot dataset are from the keyword search result pages. Even though the result page is also ranked based on video popularity, some keywords that appeared in the Livebot dataset are very unpopular, thus the videos from these topics are generally less commented. As a result, in our collected dataset, on average there are 821.85 danmu comments per video, 3.17 danmu comments per second (see Figure 3.4). In contrast, in the original Livebot dataset there are only 379.47 danmu comments per video and 2.13 per second, which is distinctly lower. For the data split of the combined dataset, we scale up the data split (2161 / 100 / 100) in the Livebot paper (Ma et al., 2019), and have 4272 / 200 / 200 videos in the training/development/test sets, respectively. Table 3.5 shows damnu statistics for the combined dataset.

We note that only selecting videos that have many danmu comments introduces a bias as unpopular videos are left out, we could try to remedy this by introducing more less-commented videos but the real unbiased assessment would be to conduct live A/B testing on the danmu video platform, this is however outside of the scope of this thesis.

We also investigate the distribution of the videos in various dimensions including the duration or the number of comments. Figure 3.6a shows the histogram of video durations, most of the videos (78%) are less than 5 mins. Figure 3.6b shows the histogram of the number of danmu comments, from it we can see that many of the videos have 500 or 1000 comments, after an investigation on the



排行榜 Leaderboard 生活 Daily life

1 说出来你们可能不信，我差点被这个小姑娘难住了 I was stumped by this little girl.
483.1万 5700 卢正义的雕刻时光

2 三句话，让医院的护士和我结了婚 I got married with a nurse with three words
182万 3343 杰克辣条

3 这才是文化膨胀！！当岩彩画遇上汉服 When rock painting meets Hanfu
138.1万 2259 造羊的岩彩学堂

4 《文艺复兴》 "Renaissance"
194.8万 9814 某幻君

5 2021第二季沙雕新闻盘点！反转反转再反转，套路套路反套路 2021 stupid news collection
160.2万 2.1万 三代鹿人

Fig. 3.4 Overviews of the 5 most popular videos in the "Daily Life" section of Bilibili. Video titles are translated on the right.

platform policy, we found that Bilibili adds a cap (500, 1000 or 3000) to the number of comments in a video based on its duration by removing the old comments.

3.3.2 Dataset Pre-processing

We follow the same procedures as in (Ma et al., 2019) to process the video frames and text content (also see Section 3.2.1). However, the acoustic information is not explored in previous work. We observe that human-created danmu comments frequently respond to speech in the video. Figure. 3.7 shows an example of this: viewers are asked in the subtitles, to post danmu comments. This motivates us to explore the potential of audio information in assisting automated commenting.

In addition, we further augment the danmu commenting dataset by extracting the audio and the subtitle information in addition to the visual and textual comment information. We follow standard practice and uniformly re-sample the audio soundtrack extracted from a video using a 16kHz standard. Then, for the audio signals, we use 20-dimensional mel-frequency cepstral coefficients (MFCCs) with 20-dimensional MFCCs derivative as audio frame features (Di Gangi et al., 2019). These are extracted with a Hanning window of 40 ms length and 32 ms hop size (32 audio vectors for each second of the audio). In order to further exploit the dataset and to strengthen the audio signal, we transcribe the

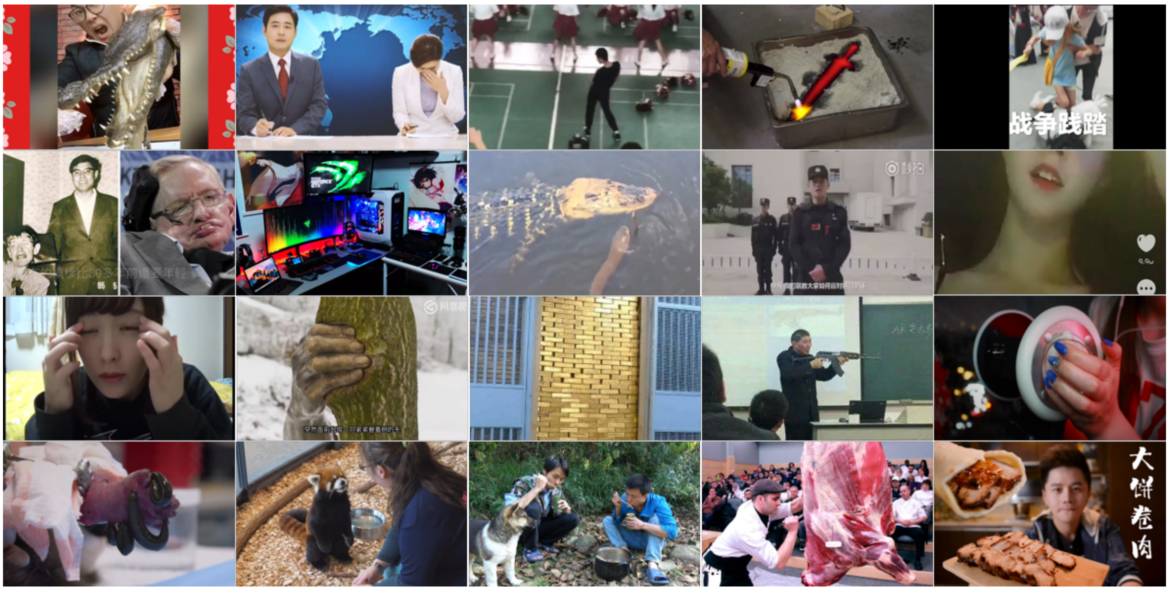
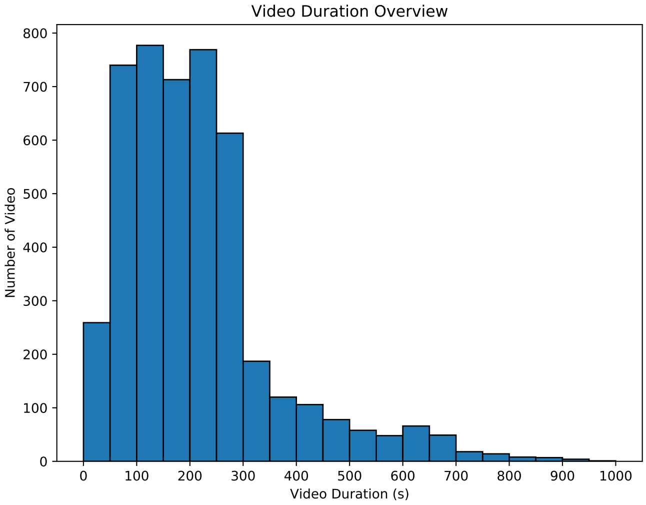


Fig. 3.5 Examples of frames from collected videos. The video content features events from daily life.

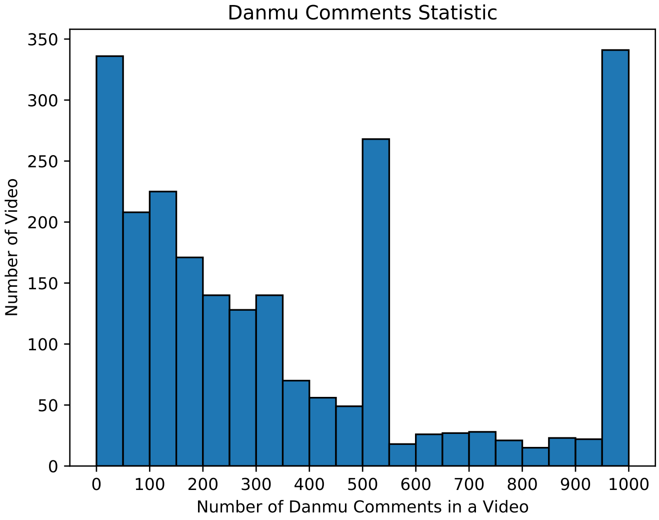
speech from the videos as additional textual resources. Instead of using speech recognition, we opt to use optical character recognition (OCR). We found that the quality of transcripts produced by speech recognition tools was by comparison of poor quality. While most of the videos on the platform embed speech subtitles that OCR tools can accurately identify. Lastly, captions also display non-speech information which could be exploited. For OCR, we use the open-source Tesseract (Kay, 2007) OCR engine on the lower half of the sampled video frames. From Figure. 3.8 we can see that for most of the videos there are less than 20 unique subtitles. Note that only 109 videos out of 4672 videos contained zero recognisable text and each video contains an average of 13.97 unique subtitles.

3.4 Conclusions

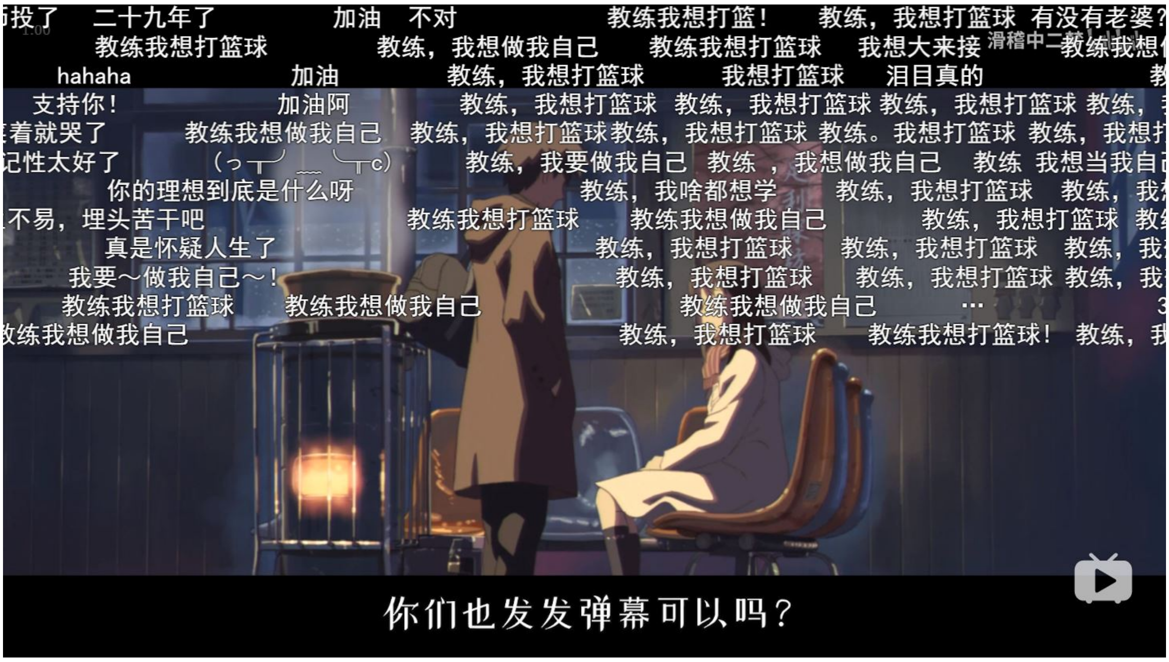
In this chapter, we began with a detailed introduction of the danmu video platform, which contained an example of the mechanism of the danmu feature and the presentation of the available data resources on Bilibili. We then examined danmu video datasets, among which we investigated the one released by Ma et al. (2019), namely Livebot. We summarized the collection details, pre-processing methods, statistics, and the issues with the released Livebot dataset. Finally, we present our expanded version of the Livebot dataset with a detailed statistical analysis. The major difference between our dataset and the Livebot dataset can be ascribed to our effort in extracting acoustic information, which includes two channels: audio soundtrack and video subtitles, the method we utilized to extract audio features is relatively rudimentary and the recent off-the-shelf speech recognition tools might improve the results, however, this is partially amended through subtitle extraction as more than 90% of the danmu video in our dataset contains subtitles. We believe our new collection will significantly contribute to our later research projects.



(a) Histogram of video duration in seconds.



(b) Histogram of the number of danmu comments.



投了 二十九年了 加油 不对 教练我想打篮！ 教练，我想打篮球 有没有老婆？
教练我想打篮球 教练，我想做我自己 教练我想打篮球 我想大来接 滑稽中二 教练我想
hahaha 加油 教练，我想打篮球 我想打篮球 泪目真的 教
支持你！ 加油阿 教练，我想打篮球 教练，我想打篮球 教练，我想打篮球 教练，
着就哭了 教练我想做我自己 教练，我想打篮球教练，我想打篮球 教练。我想打篮球 教练，我想打
记性太好了 （っ╥╯﹏╰╥c） 教练，我要做我自己 教练 ，我想做我自己 教练 我想当我自
你的理想到底是什么呀 教练，我啥都想学 教练，我想打篮球 教练，我
不易，埋头苦干吧 教练我想打篮球 教练我想做我自己 教练，我想打篮球 教
真是怀疑人生了 教练，我想打篮球 教练，我想打篮球 教练，我
我要～做我自己～！ 教练，我想打篮球 教练，我想打篮球 教练，我
教练我想打篮球 教练我想做我自己 教练我想做我自己 … 3
教练我想做我自己 教练，我想打篮球 教练我想打篮球！ 教练，我

你们也发发弹幕可以吗？

Fig. 3.7 A video frame from bilibili.com with damnu comments overlaid. The lower part of the image shows danmu comment distribution over the video. The subtitle says: “could you publish some danmu?” and the viewers are responding with a damnu burst.

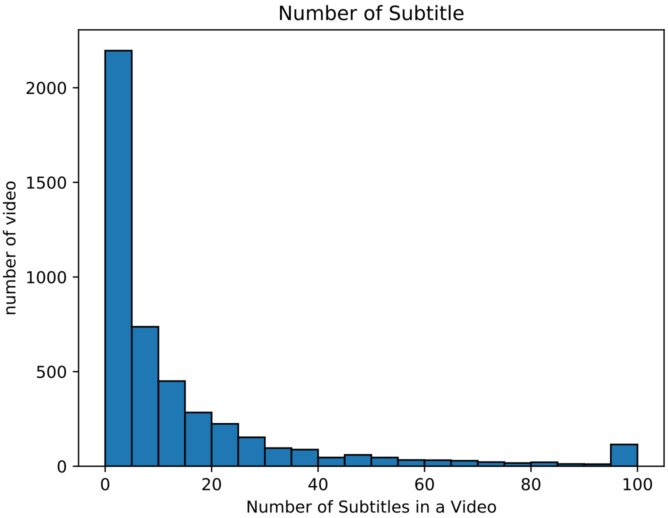


Fig. 3.8 Histogram of the number of subtitles in the videos.

Chapter 4

OpenNMT-LiveBot: A Baseline Comment Generation Platform

In the previous chapter, we introduced the danmu video dataset proposed by Ma et al. (2019) and our expanded version. Along with the danmu video dataset, Ma et al. (2019) also provides their codebase for the automated danmu commenting task. Their proposed transformer-based danmu generation module and the retrieval-based evaluation protocol have been broadly adopted in many follow-up studies, we hence choose their implementation as a starting point of our investigation.

In this chapter, we first review the related literature on the task of danmu comment generation, then uncover several issues in the Livebot codebase and report corrected scores for their experiments. In order to provide a reproducible implementation, we re-implement Livebot by using OpenNMT (Klein et al., 2017), a neural sequence learning framework, as it is well-accepted by the research community and provides a set of open-sourced APIs for language generation tasks. Our code is available on GitHub [1].

4.1 Related Work

In Chapter 2, we reviewed related technologies and methods in the field of natural language generation. In this section, we present research progress made in automated danmu commenting.

The synchronicity between danmu comments and video content makes danmu commenting a unique multi-modal application, providing rich resources for social media multi-modal analyses, such as sentiment analysis (Li et al., 2019) and video summarisation (Sun et al., 2016). Among all these applications, automated danmu commenting has become the most popular research topic in this field.

The goal of automated danmu commenting is to generate human-like comments through the analysis of the surrounding comment history and video content. This process typically involves the fusion of user comments and video streams information. When compared to other multi-modal fusion

[1] https://github.com/fireflyHunter/OpenNMT-Livebot

tasks (*e.g.* multi-modal machine translation or visual question answering, see Chapter 2), where the actual application is limited to a concrete scenario with a clearly defined goal, the scope of automated danmu commenting is not constrained by a specific scenario. Danmu comments have no explicit structure and danmu datasets used in many danmu analysis studies are usually unlabelled. Danmu comments may not actually be related to the video content. In these situations, the cross-modal connection is weak when compared to other multi-modal analysis tasks. These characteristics make automated danmu commenting extremely challenging.

The earliest work on danmu generation is perhaps the work of (Lv et al., 2019a), which proposes to directly map video frames into the comments textual space, inside a generative adversarial model. This method, however, ignores the intrinsic connections between danmu comments as they do not exploit existing nearby comments when creating new ones. A task definition with publicly available training/testing set, benchmark and baseline network architecture was proposed in Livebot (Ma et al., 2019). Their network architecture is based on the Transformer architecture and can leverage the existing danmu history as well as the visual information. This work has served as a benchmark for many follow-up studies, including (Zhang et al., 2020, Chaoqun et al., 2020), which mainly focus on improving the model architecture.

Specifically, Zhang et al. (2020) propose to apply a novel multi-modal fusion module in the danmu generation task, the proposed module builds bi-directional interactions between two input modalities. Chaoqun et al. (2020) propose to include the audio soundtrack in automated danmu commenting, their show in ablation experiments that the model gains from the addition of the audio modality. VideoIC (Wang et al., 2020b) attempts to exploit temporal information about the comment (video display timestamp) by introducing a Multi-Task decoder. Apart from predicting what the target comment should be given the context window, their architecture is alternatively trained to predict whether a target comment has been published before or after the input context window. They show that this Multi-Task approach can bring significant improvements to the danmu generation benchmark (improving mean reciprocal rank from 0.291 to 0.312 in their benchmark).

These research efforts have largely contributed to the task of automated danmu commenting and providing applicable ideas for our research goal. To begin our investigation we look at the implementation released by Ma et al. (2019) since it provides a complete benchmark for automated danmu commenting, including dataset, and baseline implementation, as well as an evaluation protocol.

4.2 Livebot Implementation

In this section, we present an overview of the danmu comment generation framework of the Livebot and report several issues we identified when reviewing the released code by the Livebot team. We address these issues and report the updated results at the end of the section.

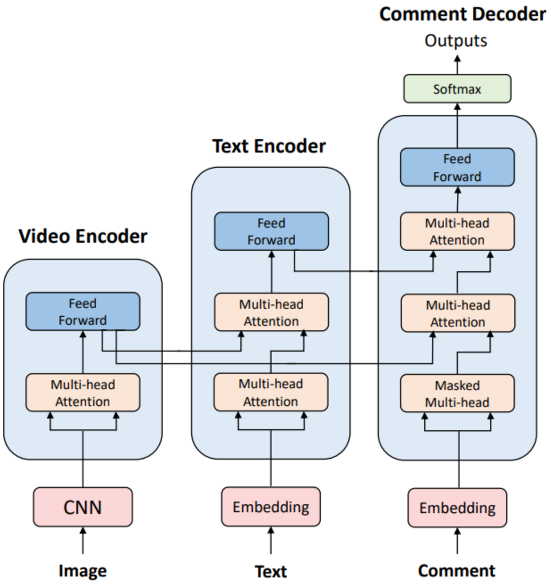


Fig. 4.1 Structure of the Transformer model in (Ma et al., 2019), the encoding CNN is a pre-trained resnet18.

4.2.1 Task Definition

The danmu comment generation task is defined in Livebot (Ma et al., 2019) as follows. Given a video V, a video display timestamp t, and the surrounding comments C near the timestamp, an AI agent aims to generate a comment y relevant to the clips or the other comments near the timestamp t.

4.2.2 Model Architecture

We follow the model structure described in (Ma et al., 2019), and illustrated in Figure 4.1 (see section “Model II: Unified Transformer Model” in (Ma et al., 2019)). Comments and video frames are encoded using a Transformer architecture (Vaswani et al., 2017). The model consists of three parts: the video encoder which encodes video frames into a visual representation; the text encoder which generates the contextual vector by encoding the sequence of input words combined with the visual representation; and finally the comment encodes which combines these vectors in a comment decoder to generate output tokens recursively.

Video Encoder The video encoder first encodes each frame I_i in the current analysis window with a pre-trained CNN (ResNet-18). Then it uses a Transformer layer to encode the sequence of visual vectors into a visual representation h:

$$v = ResNet18(I) \tag{4.1}$$

$$h = MultiHeadAttention(v, v, v) \tag{4.2}$$

Text Encoder Comments from C are concatenated into a word sequence e as the input of the text encoder. The attention-aware representation of textual input e can be stated as:

$$g = MultiHeadAttention(e,h,h) \tag{4.3}$$

In this Transformer module, there are two multi-head attention components where the first one attends to the text input e and the second attends to the outputs of the video encoder h.

Comment Decoder A Transformer module with three multi-head attention units is applied in the comment decoder, where the first one attends to the comment input y and the last two attend to the outputs of video encoder h and text encoder g respectively.

4.3 Evaluation Metrics

The common reference-based NLG evaluation metrics such as BLEU Papineni et al. (2002), ROUGE Lin (2004) turn out not to be well suited to the Damnu comments generation task. This is because the content of the danmu comments at a particular video display timestamp can be very diverse, while these metrics typically require a unique topic. Das et al. (2017) were the first to acknowledge this and proposed a retrieval based evaluation protocol for visual dialogue, their application is about generating dialogues from video content. Ma et al. (2019) then proposed to also adopt a retrieval-based ranking evaluation plan in measuring the performance of a danmu commenting system: a candidate comment set is constructed for each test sample, then the model is asked to sort the candidate set; it is assumed in (Das et al., 2017, Ma et al., 2019), that a good model can rank the correct comments at the top of the proposed comment set.

The candidate comment set for each test sample contains 100 comments of four different types:

- **Correct**: five groundtruth comments from humans.
- **Plausible**: twenty comments most similar to the title of the video based on tf-idf score.
- **Popular**: twenty most frequent comments in the training set.
- **Random**: random comments are taken from the training set to ensure there are 100 unique comments in the generated output set.

Retrieval-based evaluation metrics are used in the reported experiments to automatically evaluate the commenting system, we follow (Ma et al., 2019) and use the following retrieval metrics to evaluate the results:

- **Recall@k**: the proportion of human comments found in the top-k recommendations.

Model	Fluency	Relevance	Correctness
Transformer	4.31	3.07	3.45
Human	4.82	3.31	4.11

Table 4.1 Human evaluation results on the Livebot test set, as reported in (Ma et al., 2019) (higher is better).

- **Mean Rank(MR)**: the mean rank of the human comments.

$$MR = \frac{1}{N}\sum_{i=1}^{N} rank_i \tag{4.4}$$

- **Mean Reciprocal Rank(MRR)**: the mean reciprocal rank of the human comments.

$$MRR = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{rank_i} \tag{4.5}$$

We also report the confidence interval for each of these metrics. For recall@k we use the confidence interval for population proportions with a confidence level at 95%. For MR and MRR, we use the confidence interval with the same confidence level.

4.4 Human Evaluation

Apart from the automatic ranking evaluation process, (Ma et al., 2019) also proposes to include human judgements in evaluation. In their paper, the generated comments are evaluated, by three native mandrain speakers (No further demographic information about the evaluators are provided), in three aspects: **Fluency**, **Relevance** (whether the generated comments are related to the video content) and **Correctness** (how close are the generated comments to human comments). The reference comments in the test set are also evaluated. The results of human evaluation are presented in Table 4.1. This shows that the comments generated from Transformer are close to those of real-world danmu comments in terms of relevance and fluency.

4.5 Identified Issues in Reproducing the Livebot Results

We first tried to reproduce the work of Livebot (Ma et al., 2019) using the released code and the dataset. For reference, the Livebot results are reported in Table 4.2 and labelled with "Livebot paper". Specifically, results with different input are reported (*e.g.* "Text Only" means text input are all masked during evaluation). We conducted our experiments using the code provided on the authors' Github project page and used the same model structure and configurations (batch size, learning rate etc.) described in (Ma et al., 2019). The results we obtained are shown in the same table with the label

Input	Dataset	Run Label	Recall@1	Recall@5	Recall@10	MR	MRR
Text Only	unknown	Livebot paper	13.95	34.57	51.57	17.01	0.251
	Livebot Github	Issue #1	5.41 ± 0.05	20.33 ± 0.18	34.58 ± 0.31	23.78 ± 0.44	0.147 ± 0.01
	Livebot Github	Issue #1-2	9.77 ± 0.87	24.31 ± 0.22	31.15 ± 0.28	21.10 ± 0.51	0.185 ± 0.008
	Livebot Github	Issue #1-3	17.11 ± 0.16	37.07 ± 0.35	51.08 ± 0.48	14.91 ± 0.48	0.280 ± 0.009
	No duplicate	Issue #1-4	12.04 ± 0.11	25.01 ± 0.23	42.04 ± 0.40	20.77 ± 0.65	0.219 ± 0.01
	No duplicate	OpenNMT	12.48 ± 0.12	24.16 ± 0.22	42.68 ± 0.41	18.66 ± 0.49	0.228 ± 0.01
	Expanded Dataset	OpenNMT	147.8 ± 0.18	34.13 ± 0.31	47.89 ± 0.41	18.11 ± 0.22	0.277 ± 0.03
Visual Only	unknown	Livebot paper	11.40	32.62	50.47	18.12	0.231
	Livebot Github	Issue #1	5.44 ± 0.05	20.30 ± 0.19	36.31 ± 0.35	23.84 ± 0.45	0.142 ± 0.01
	Livebot Github	Issue #1-2	8.66 ± 0.08	22.64 ± 0.21	31.42 ± 0.30	21.23 ± 0.54	0.175 ± 0.009
	Livebot Github	Issue #1-3	7.78 ± 0.07	26.78 ± 0.25	40.23 ± 0.38	19.66 ± 0.54	0.183 ± 0.01
	No duplicate	Issue #1-4	6.55 ± 0.06	23.41 ± 0.22	39.38 ± 0.38	20.77 ± 0.67	0.169 ± 0.01
	No duplicate	OpenNMT	7.01 ± 0.06	24.35 ± 0.23	37.76 ± 0.36	19.89 ± 0.43	0.172 ± 0.01
	Expanded Dataset	OpenNMT	8.21 ± 0.12	27.08 ± 0.29	46.80 ± 0.64	19.07 ± 0.28	0.187 ± 0.02
Text+Visual	unknown	Livebot paper	18.01	38.12	55.78	16.01	0.275
	Livebot Github	Issue #1	5.81 ± 0.05	21.49 ± 0.19	36.43 ± 0.35	22.22 ± 0.45	0.155 ± 0.01
	Livebot Github	Issue #1-2	11.46 ± 0.11	26.22 ± 0.24	32.96 ± 0.29	19.54 ± 0.48	0.204 ± 0.009
	Livebot Github	GitHub Issue	10.56	25.24	34.05	20.26	0.170
	Livebot Github	Issue #1-3	18.79 ± 0.17	39.46 ± 0.38	50.13 ± 0.48	16.17 ± 0.46	0.297 ± 0.01
	No duplicate	Issue #1-4	15.50 ± 0.14	34.57 ± 0.33	48.48 ± 0.46	17.25 ± 0.48	0.260 ± 0.01
	No duplicate	OpenNMT	14.79 ± 0.14	33.45 ± 0.32	48.93 ± 0.46	17.45 ± 0.49	0.257 ± 0.01
	Expanded Dataset	OpenNMT	18.83 ± 0.16	34.50 ± 0.33	52.17 ± 0.51	17.81 ± 0.36	0.347 ± 0.003

Table 4.2 Objective Metric Results. “Run Label” column refers to the version of the experiment, *e.g.* Issue #1-3 means the experiments with issue 3 and all past issues fixed. "Livebot paper" refers to the results reported in (Ma et al., 2019). Recall@k, MRR: higher is better; MR: lower is better.

“Issue #1”. Clearly, the results from our experiments are much lower than the baselines. In order to explore the reasons for the performance mismatch, we conducted a series of investigations examining the GitHub implementation and the released dataset. From our investigation, we identified several issues with GitHub implementation which are presented below.

4.5.1 Issue #1: Candidate Set Ranking

First, in the implementation, the re-ranked candidate list is sorted based on the cross-entropy loss in descending order. However, according to the paper, a good candidate should be placed at the top side of the candidate list, in this case, the cross-entropy loss should be sorted in ascending order. This issue is also raised in the GitHub issue page by another researcher [2], the corresponding results are labelled ‘GitHub Issue” in Table 4.2. We report the results with this issue fixed (see “Issue #1-2”). The scores are very close to the results from the GitHub issue page, we can see that after fixing the ranking problem the scores improve a little, but are still significantly lower than the reported Livebot baselines.

[2] https://github.com/lancopku/Livebot/issues/1

4.5.2 Issue #2: Candidate Scores

We then carefully looked at the evaluation code and noticed a subtle error in the candidate score computation: in the original implementation the score of a candidate is computed as the sum of the cross-entropy loss for every token rather than the mean value. This results in an advantage for short candidates and, in fact, we found that the top re-ranked positions in the list are mostly occupied by comments of only one word.

We corrected the code by averaging the score over every non-ignored token (tokens for padding and separating are ignored when computing cross-entropy loss). Thus instead of

$$\text{Score}(c) = \sum_{i=0}^{L} \text{CrossEntropy}(g_i, hi)\,, \tag{4.6}$$

we implemented:

$$\text{Score}(c) = \frac{\sum_{i=0}^{L} \text{CrossEntropy}(g_i, h_i)}{\#\text{Valids}}\,, \tag{4.7}$$

where g_i and h_i are the i-th output token and groundtruth token, L is the maximum length of the model output (including padding), #Valids is the number of valid tokens in a candidate. The results reported as “Issue #3” in Table 4.2, at this step we obtain scores that are closer to the baselines.

4.5.3 Issue #3: Construction of the Plausible Set

We also found an inconsistency in constructing the plausible set. It is described in the paper that when building the candidate list, the plausible set is retrieved based on the video title. However, in the implementation, we noticed that the plausible set is retrieved using current context comments (The comments surrounding the groundtruth comment, which is also the text input) as the query rather than the video title. Unfortunately, the mapping between the raw dataset and the provided dataset are not given, so we are not able to reconstruct the provided dataset from the raw dataset, and hence could not directly compare the results with and without fixing this issue. In the experiments (“Issue #1-4”) reported in the next section, we follow the Livebot paper and use the video title to construct the plausible set.

4.5.4 Issue #4: Training and Testing Sets

As we report in Section 3.2.2, 38 videos appear twice in the original Livebot dataset, this redundancy will affect the model performance. In order to investigate the model performance in the normal scenario, we remove the redundant videos from the training set and conduct experiments on the updated Livebot dataset. This dataset is labelled as “No duplicate” in the result table. The results after removing the duplicate videos are shown as “Issue #1-4” in Table 4.2. Compared to “Issue #1-3” the performance can be observed to be slightly lower, which is what we could anticipate since the model no longer gains from the overlapped information across the training and test set.

4.6 Re-Implementation Using OpenNMT

In order to provide a reproducible implementation for later research on the danmu commenting task, we re-implemented the Transformer network of LiveBot using the OpenNMT (Klein et al., 2017) open-source neural machine translation framework. Since this provides highly configurable model architectures and training procedures and is widely adopted by the research community.

The vocabulary size is set to 30,000 to keep it consistent with the original paper, and in Transformer architecture, the size of the word embedding and hidden layer are set to 512, as in (Ma et al., 2019). Additionally, the batch size is set to 64 and the dropout rate to 0.2. The optimization method is chosen as Adam (Kingma and Ba, 2014), with $\beta_1 = 0.9$ and $\beta_2 = 0.998$.

4.7 Experiments with OpenNMT-Livebot

Results of this re-implementation are reported in Table 4.2 under the label "Re-implementation". With past issues resolved. At this stage, the scores we get are very close to run "Issue #1-4"

We also conduct experiments on the expanded dataset, which are labelled "Expanded" in the result table. Realising that scores on the expanded dataset are not directly comparable to the Livebot results since both the training and the test sets have doubled their sizes, we observe a significant increment for scores of the expanded dataset over Livebot scores. Recall that the danmu comment densities of our additional collected data are much higher than those in the original Livebot dataset. This means on average, there are potentially more available textual contexts when generating new comments, and this could be the reason underlying the improvement of scores.

4.8 Conclusions

In this chapter, we discussed related work in automated danmu commenting. Then, the code presented as the official LiveBot implementation was reviewed, in this process, we identified several discrepancies with the original paper. We addressed each of these issues and reported updated results accordingly. We also introduced a new baseline implementation using the OpenNMT framework. The updated baseline results are still lower than the ones reported in the original Livebot paper. However, since we do not have the access to the exact version of the code used to produce the original results, we are not able to determine the exact reasons for these differences. Additionally, we did not compare the results to the two derivatives work of Livebot (Zhang et al., 2020, Chaoqun et al., 2020) since we are concerned that their work might be impacted by what we uncovered in Section 4.5. As their code hasn't been published, we choose not to rely on their results. Based on our experiments and analysis, we believe this performance gap is caused by the removal of duplicate videos. We further measure the performance of the OpenNMT framework on our expanded dataset. We believe the implementation and scores generated are valid and should serve as a suitable baseline for our later studies.

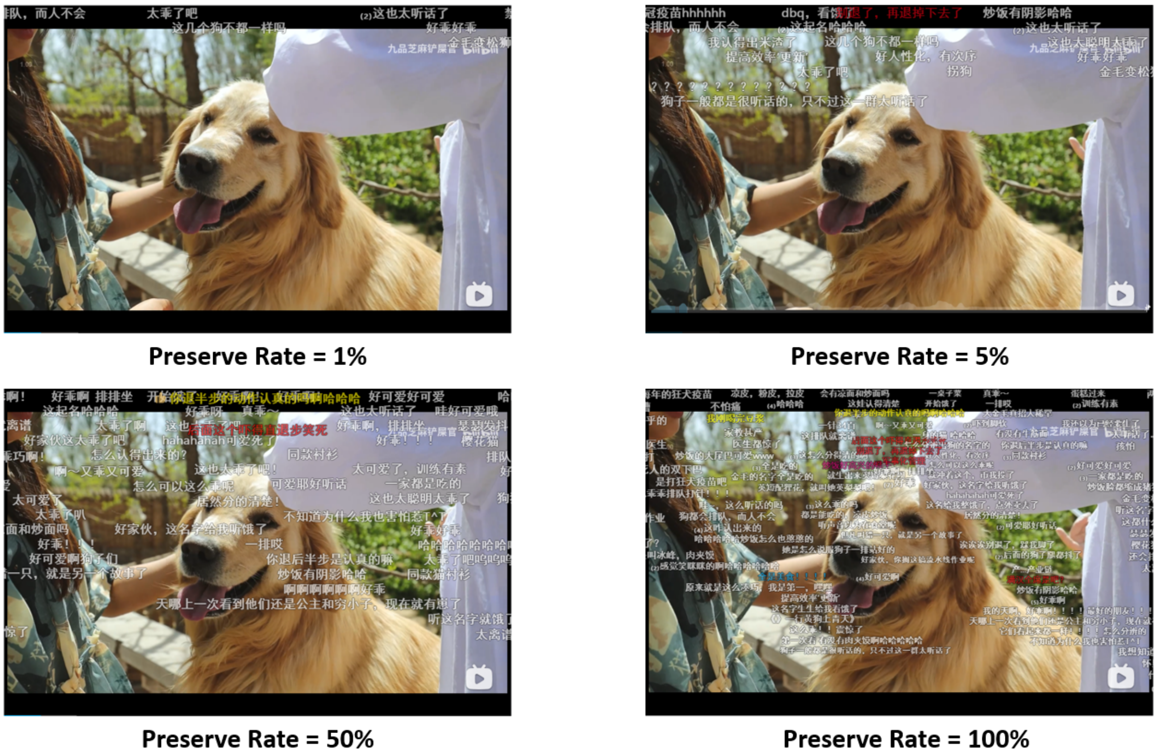
Chapter 5

The Cold Video Problem in Danmu Comment Generation

In this chapter, we address our first research question ([RQ1] Can we automatically create meaningful comments for less-commented or even uncommented videos?) by investigating the cold start scenario in danmu comment generation.

As the most representative feature of the danmu video platform, danmu comments can to some extent affect user's viewing experience (Yang, 2018). The number of danmu comments has become a key indicator in danmu video platforms. In such a scenario, videos with many danmu comments stand a higher chance of being recommended or searched and naturally attract more viewers. Most of the existing literature (Ma et al., 2019, Wang et al., 2020b, Chaoqun et al., 2020, Zhang et al., 2020, Lv et al., 2019a) mentioned in the previous chapter has focused on the analysis of videos that already have many comments. This is however probably not the most critical scenario for automated danmu generation as these videos are already popular. Also, it is easier in these cases to exploit the numerous nearby comments to generate new comments. Similar to the "cold start problem" in recommender systems, the real issue faced by content creators is that videos need many danmu comments to start attracting traffic.

In this chapter, we propose to solve this "video cold start problem" by a method that can generate danmu comments on videos that have zero, a few, or many comments. We propose a multi-density cold video Transformer (MCVT) that can leverage multi-modal signals including surrounding comments, video frames, but also subtitles and audio signals in an end-to-end neural network. The key idea is then to approach the task globally and train the network for different comment density scenarios (see Section 5.3). To achieve this, we leverage the video publication date of comments from the video platform. This allows us to consider different snapshots of a video's commenting lifetime (ie. when the video was freshly uploaded with no comments, then when it had a few comments, and later with many comments, see Figure 3.7 for a detailed illustration). This information has not been exploited in the existing work of danmu comment generation, but we show that it can be used effectively in the process of danmu generation.


Preserve Rate = 1%

Preserve Rate = 5%

Preserve Rate = 50%

Preserve Rate = 100%

Fig. 5.1 Screen shots of the different commenting lifetime of a Bilibili video, 1% preserve rate means only the earliest one percent danmu comments published are preserved and presented in the video playback screen.

We conduct the experiments on our expanded danmu video dataset(see Section 3.3) and evaluate our system in Section 5.4 through both a retrieval-based evaluation method and human judgement. Results show that our system can produce comments that are close to the quality of human comments.

To make our work fully reproducible, we have our source code made publicly available. [1] The work described in this chapter is published in NAACL Multimodal Artificial Intelligence workshop (Wu et al., 2021b).

5.1 Task Overview

In order to address the “video cold start problem”, we propose a novel task that requires the automated commenting system to generate danmu comments given danmu videos at different commenting stages. In this section, we define our proposed task and describe the video content extraction methods used in this investigation.

[1] https://github.com/fireflyHunter/Cold-Video-Danmu-Generation

5.1.1 Task Definition

To address the cold start problem we aim to generate high-quality comments given videos with different comment densities. In order to handle different danmu density scenarios of the cold start problem, we first sort the existing comments $\mathbf{C}$ for a video by their publication date and only keep a subset $\mathbf{C}_p$ consisting of a percentage p of the earliest comments of the video. This strategy is enforced to reconstruct video danmu comments in different phases of their lifetime as we illustrate in Figure 3.7. Then we define our task as follows: given a video $\mathbf{V} = \{\mathbf{s_0}, \ldots, \mathbf{s_L}\}$ (following accepted convention $\mathbf{V}$ is split into segments of one second duration (Ma et al., 2019, Wang et al., 2020b)), the generation module is asked to generate a target comment $\boldsymbol{y}$ using comments from $\mathbf{C}_p$ and the k previous seconds of the video clip $\mathbf{s_{[i-k,i]}}$.

5.1.2 Dataset Formation

We use our expanded danmu video dataset described in Chapter 3.3 for the experiments. The whole dataset is split into training/dev/test set with 4272, 200 and 200 videos respectively.

The training data for a particular level p, percentage of existing manual comments preserved, is defined as follows. Each target comment for the training set is randomly sampled from the original comment set $\mathbf{C}$ and the corresponding comment's context is defined as the 5 nearest comments from $\mathbf{C}_p$ that precede the target danmu in the video timeline. This follows the observation made in Livebot (Ma et al., 2019), that the semantic and textual similarity of comments is correlated to their timeline proximity and that the danmu context should be limited to the 5 nearest comments. We also add a *causality* constraint by applying the constraint that the comments must have been published before the target danmu in natural time.

We sample the training data for $p = 0\%, 5\%, 30\%, 50\%, 70\%$ and 100%, to form a combined training set of 4,800,145 pairs of target comment/context comments. Target comments can be sampled multiple times for different contexts.

For the 200 videos of the test set, we focus on the video highlights by only selecting 1,879 comments in the most frequently commented moments in the video timeline. To study the system performance under different comment densities, we build one test set for each of the proposed values of p.

5.2 Network Architecture

Our proposed model, presented in Fig 5.2, applies standard Transformer modules with an encoder-decoder architecture.

During the encoding stage, visual, audio and text features are first encoded respectively, then three Transformer modules are used to fuse the information for the three modalities recursively. In the decoder, the target comment is decoded through a Transformer layer with multiple multi-head attention modules that attend to three encoded multi-modal representations respectively.

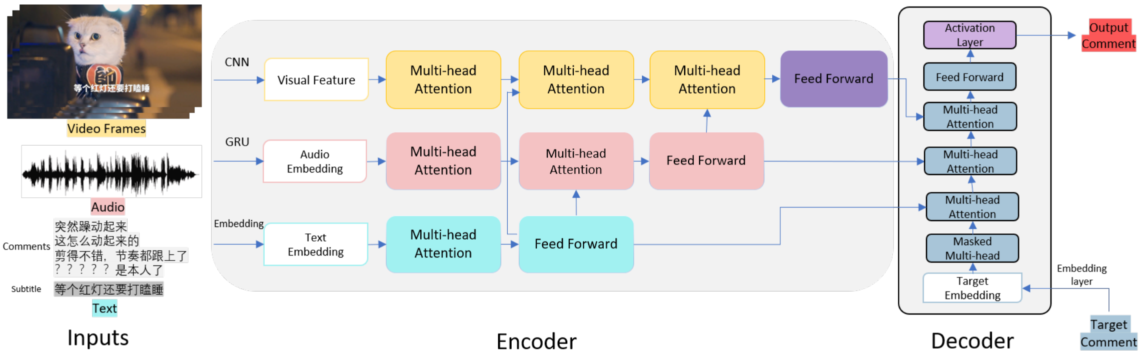


Fig. 5.2 Architecture of our proposed model.

5.2.1 Video Encoder

As described in Section 3.3, video frames are encoded through a pre-trained 18-layer ResNet. We take the output from the last pooling layer of ResNet as the visual feature, the frame vector of the i-th second of the video is denoted as $v_i \in \mathbb{R}^{n_{18}}$, where $n_{18} = 512$ is the size of the resulting ResNet18 features. The frame vectors in the video clip are combined as $\hat{v}_i = \{v_{i-k}, \ldots, v_i\}$.

5.2.2 Audio Encoder

For the audio signal, we use 20-dimensional mel-frequency cepstral coefficients (MFCCs) with 20-dimensional MFCCs derivative as audio frame features (Di Gangi et al., 2019). These are extracted with a Hanning window of 40 ms length and 32 ms hop size. We include all audio frames as the audio input, hence we sample 32 audio vectors for each second of the audio. The audio information at time point i is denoted as a_i^j, where j is the j-th audio frame vector in the window analysis at time i. A GRU module (Cho et al., 2014) is applied to recursively encode the input audio sequence. At each stage, the current hidden state h_i^j is calculated based on the last hidden state h_i^{j-1} and the current input audio frame vector a_i^j. The sequence of hidden states h_i^j for all audio frames is concatenated into an audio encoder output $\hat{a}_i \in \mathbb{R}^{n_a \times 512}$, where $n_a = 32 \times k$ is the number of audio frames in the analysis window and 512 the dimension of the hidden state.

5.2.3 Text Encoder

Contextual comments are concatenated with a special delimiter token T_d inbetween each comment and then combined with the unique subtitles from the analyzed k second window. As opposed to Livebot (Ma et al., 2019), where there are always 5 context comments, in our cold start scenario we sometimes have less than 5 and even 0 comments. In the extreme case, we use a special token T_n with an empty comment field to show that no context comments are available.

All unique subtitles within the analysis window $\mathbf{s_{[i-k,i]}}$ are also concatenated with the same delimiter token. Finally, we form the text input by combining comment sequence and subtitle sequence with T_d.

We remove the punctuation and segment words using Jieba [2](a popular Chinese word segmentation tool). Each word of text input is then passed to an embedding layer of size $d \times |V|$, where d is the dimension of the word embedding and $|V|$ is the size of the vocabulary. After embedding, the text input for analysis window $\mathbf{s_{[i-k,i]}}$, is now represented as $\hat{e}_i \in \mathbb{R}^{n\times d}$.

5.2.4 Fusion of Modalities

Following the success of the Transformer architecture in multi-modal processing (Ma et al., 2019, Chaoqun et al., 2020), we adopt a multi-unit Transformer module to recursively learn and combine representations from all three modalities. The transformer unit first encodes the text input $\hat{e}_i$ into a transitional hidden state H_e as we believe textual inputs provide most of the contextual information in process of multi-modal video scene analysis: text is proved in the ablation study of Livebot to be most contributed modality and we observe that danmu comments frequently interact with previous comments.

$$H_e = MultiHeadAttention(\hat{e}_i, \hat{e}_i, \hat{e}_i)$$

Then, a second transformer unit combines H_e and the input audio with two multi-head attention modules, the first one attending to $\hat{a}_i$ and the second one attending to H_e.

$$\begin{aligned} H_a &= MultiHeadAttention(\hat{a}_i, \hat{a}_i, \hat{a}_i) \\ H_{ae} &= MultiHeadAttention(H_a, H_e, H_e) \end{aligned}$$

Finally, another unit with three multi-head attention modules is used to summarise the video clip representation H_{vae}.

$$\begin{aligned} H_v &= MultiHeadAttention(\hat{v}_i, \hat{v}_i, \hat{v}_i) \\ H_{ve} &= MultiHeadAttention(H_v, H_e, H_e) \\ H_{vae} &= MultiHeadAttention(H_{ve}, H_{ae}, H_{ae}) \end{aligned}$$

5.2.5 Decoder

In the model decoder, the output comment is generated through a Transformer layer with 4 multi-head attention modules that attend to the target comment $\boldsymbol{y}$, text hidden state H_e, visual hidden state H_{ae} and audio hidden state H_{vae} respectively. Then the probability of the output comment is produced with a softmax layer on top of the decoder output.

[2] https://github.com/fxsjy/jieba

5.3 Network Training Regime

In this section, we introduce our proposed multi-density learning strategy which is designed to cope with the diversity of the training set. We also present detailed training information for the system.

5.3.1 Multi-Density Learning

Our method considers all the different cold start scenarios together, thus the text inputs of the proposed model can be very different during the training of the system. In the extreme case, the text input can be empty when there happen to be no available danmu comments (complete cold start scenario) and subtitles.

We adopt a multi-task training strategy to handle all the cold start scenarios at once. In detail, our training regime is implemented by randomly assigning, at each mini-batch, the percentage p of earlier comments that are kept from a fixed set of values $\{0\%, 5\%, 30\%, 50\%, 70\%, 100\%\}$. Recall that $p = 0\%$ corresponds to the cold start problem, and $p = 100\%$ corresponds to the situation where all other comments are available (such as in Livebot (Ma et al., 2019)). By alternating between these values of p, we are able to train the network for both the cold start and Livebot scenario.

5.3.2 Training Details

The video analysis window size k is set to 5 (s). For the text input, we build the vocabulary by selecting the most frequent 50,000 words in the dataset and set the max length of the input text sequence to 100. In the model, the text embedding is of size 512 and is randomly initialized before training. The dimension of the audio's GRU hidden state is set to 512. We apply the same setting for all Transformer components used in the network. For each Transformer, the hidden state dimension is set to 512, the feed-forward network dimension is 2048, the number of heads is 8 and the number of blocks is 6. The loss criterion is cross-entropy. The number of epochs is set to 10, the batch size to 64 and we use the Adam optimizer (Kingma and Ba, 2014) with settings β_1 = 0.9, β_2 = 0.998, weight decay =$1 \times 10^{-4}, \varepsilon = 1 \times 10^{-8}$ and learning rate 1×10^{-4}. All training was done on a Linux server with a single RTX 2080 Ti graphic card, 16 cores Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz and 256GB RAM. The model is implemented using Pytorch 1.4.0 and Python 3.6. With the above settings, it takes around 34 hours to complete the training.

5.4 Experimental Investigation

In this section, we report results for our investigation of comment generation. We use the Livebot model (Ma et al., 2019) as a baseline. Specifically, we use our code from (Wu et al., 2020b), trained on our full dataset with only video frames and surrounding comments as input. The models proposed in (Chaoqun et al., 2020, Zhang et al., 2020) are very recent and their code is not publicly available, so we do not consider these as one of our baseline methods. Other older neural architectures such as

LSTM are also not included in this study since it is well established that Transformers are the method of choice for modelling multi-modal signals.

5.4.1 Scope of Experimental Study

In these experiments, we evaluate the effectiveness of our proposed system with particular attention to the following three aspects:

- The benefits of added audio and subtitle features.
- The benefit of the multi-density training regime.
- System performance in the cold start scenario.

In order to examine the advantages of using added information, we compare the results of our model against the Livebot baseline. Then, we compare the model trained solely on the single cold start scenario with the one applying the multi-density training regime to evaluate the effectiveness of our training strategy. To specifically investigate the complete cold start scenario, we experiment by having our system trained uniquely on the complete cold start cases for $p = 0\%$.

In the following subsection, we describe the evaluation metrics used in these experiments and then present our experimental results.

5.4.2 Evaluation Metrics

Reference-Based Metrics

As discussed earlier in Section 4.3, standard NLG metrics such as BLEU and ROUGE are not suitable for our task. Instead, following Das et al. (2017), we evaluate our system through a retrieval based protocol by asking the model to re-rank a candidate set which consists of five ground truth comments and 95 irrelevant comments.

We report as in Section 4.3 the Recall@k, Precision@k, Mean Rank (MR) and Mean Reciprocal Rank (MRR) along with their confidence interval scores as evaluation metrics.

Human Evaluation

Human judgments can help obtain a more intuitive and reliable measurement of the generated comments. Thus, similarly to Livebot, (see section 4.3) we conducted a human evaluation to further evaluate our system.

A subset of 50 videos was randomly sampled from the 200 videos of the test set. Three native Mandarin speakers (all between 23 and 27 years old, consisting of two males and one female) familiar with danmu were asked to rate the quality of the generated comments on three criteria: fluency, relevancy and engagement.

- **Fluency** is intended to measure the language quality of the generated comment.

- **Relevancy** measures the semantic relevancy between the generated comment and the input video and nearby comments.
- **Engagement** should reflect how likely it is that the generated comment will motivate others to respond.

As proposed in Livebot, we measure **Fluency** and **Relevancy**. However, we replace **correctness** with **engagement** as we believe it is more relevant to our overall objective of engaging human viewers into danmu conversation.

The score for all 3 measurements ranges from 1 (poor) to 5 (excellent). The final score is the average of the scores of the three annotators. The evaluation was conducted on the comments generated by our method for $p \in \{0\%, 5\%, 50\%\}$. For reference, we also evaluate the groundtruth comment set for these videos.

5.4.3 Experiments and Results

In this ablation study, we compare 4 variants of the model.

- **Livebot** (Ma et al., 2019) leverages textual and visual information in a Transformer architecture. This is trained on the extended dataset using our Open-NMT implementation. The training is done here with p=100%.
- **Livebot-t** applies the same network architecture as **Livebot**, but is trained with our multi-density training strategy to evaluate the effectiveness of our proposed training regime.
- **MCVT** is the final system proposed in this work, which includes the training regime and the inclusion of the additional audio and subtitle features.
- **MCVT-Zero** is listed to further examine the performance limit in the cold start scenario, i.e. we assume a situation where no comments are present. Thus, we train the MCVT network uniquely on the cold start scenario for $p = 0\%$

We do not include Chaoqun et al. (2020) and Zhang et al. (2020) as a baseline method for the same concern we mentioned in Section 4.8.

The retrieval task results are reported in Table 5.3 and Figure 5.3. The results in Table 5.3 show that **Livebot-t** outperforms the baseline **Livebot** model in most cases, and thus demonstrates the effectiveness of our training regime. One exception is found when $p = 100\%$, the **Livebot** model, trained only with densely commented videos, slightly outscores **Livebot-t**, we think this means the information learned from multi-density training strategy produces extra noise when the model only aims to generate comments for popular videos. By contrast, from the third and fourth rows of Table 5.3, we can see that our **MCVT** model has similar performance to **MCVT-zero**, which has been trained specifically for the complete cold start scenario. In this situation, the extra knowledge gained from learning popular videos does not appear to affect the performance in the cold start situation. This

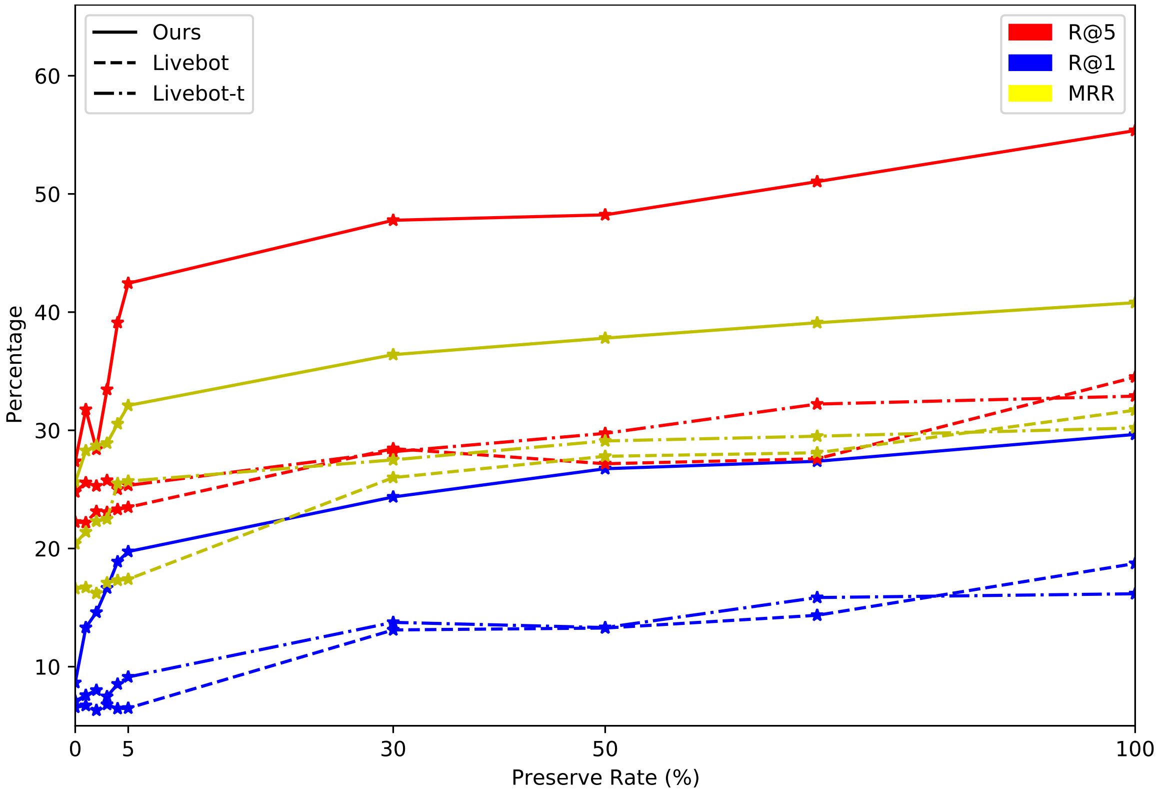


Fig. 5.3 Model performance for R@5, R@1, MRR, at different comment densities p (see Table 5.3).

comparison between the behaviour of the **Livebot** and **MCVT** systems potentially demonstrates the advantage of our multi-density training regime in the case of the cold start scenario.

We also see that our model outperforms **Livebot-t** in every scenario, which also supports the idea that integrating the audio signal and subtitle in the generation system can significantly improve the performance of the model.

Table 5.1 reports the results of the human evaluation. We can see that our proposed architecture (MCVT) systematically outperforms the baseline method (Livebot) in each criterion. The overall performance of the model is almost indistinguishable from real danmu comments. Our relevancy and engagement scores are higher when $p \geq 50\%$. The quality of our model degrades slightly for the complete cold start scenario, but the results are still quite close to human comments.

5.4.4 Case Study

Examples of predicted outputs are shown in Fig. 5.4. In the first case, the corresponding video frame shows a groundhog being fed. The subtitle and the generated comments are reported in the table to the right. We can see that the model generates reasonable comments, which are relevant to the video shot and match the video's positive emotion (*e.g.* "laugh", "hahaha" and "lol"), even in the case of a completely cold start.

Model	p	Fluency	Relevance	Engagement
Livebot	0%	3.88	2.58	2.55
MCVT	0%	4.25	3.17	2.76
Livebot	5%	3.89	2.51	2.31
MCVT	5%	4.33	3.36	2.99
Livebot	50%	4.01	3.17	2.78
MCVT	50%	4.59	3.78	**3.07**
Livebot	100%	4.30	3.22	2.81
MCVT	100%	4.47	**3.91**	2.97
Human	-	**4.79**	3.58	3.01

Table 5.1 Human evaluation on 50 videos from the test set. Each comment is graded between 1 and 5, by 3 reviewers, for their language fluency, relevance to the video content, and how likely they are to provoke other viewers to also comment.

5.5 Conclusions

In this chapter, we aimed to answer the first research question and investigated the cold video start problem in the danmu video platform. This is motivated by the fact that the number of danmu comments is a determining factor of video popularity. We leveraged this finding in automated danmu comment generation and aimed to generate comments from all types of videos including less popular ones. We proposed a multi-modal fusion network that includes the processing of video frames, already published comments, and also audio and caption text to solve this task. In order to handle different comment density scenarios in the dataset, we adopted a multi-density learning strategy during training of the system and performed extensive experiments on an expanded danmu video dataset. Results demonstrate the advantage of our method over the state-of-the-art in solving the cold video start problem. This supports the ideas of integrating subtitles and audio soundtracks as well as adopting the multi-density training regime.

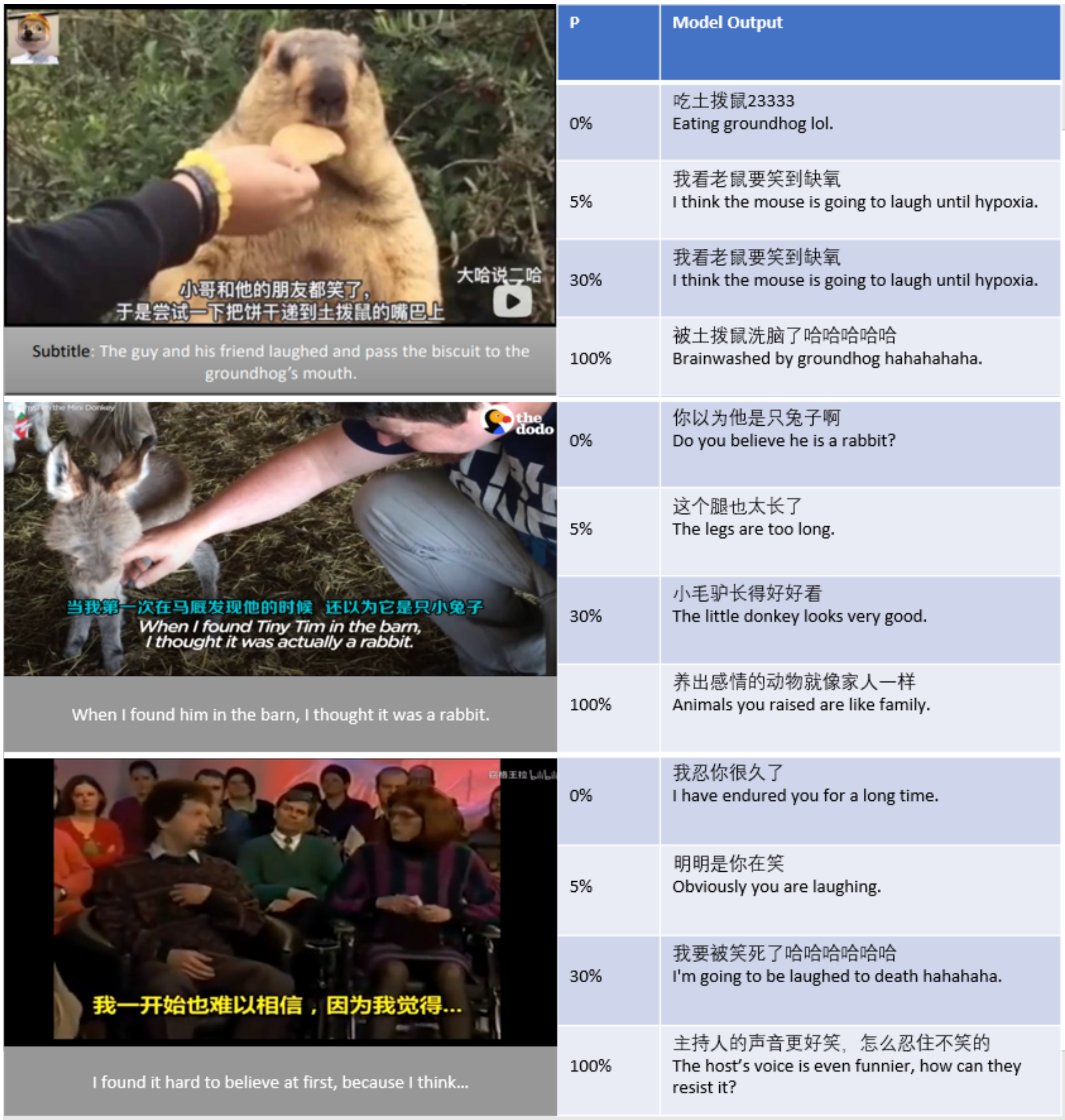
At this stage, we are able to generate high-quality human-like comments from videos with zero, little or many comments. To further improve the system, our next research goal will be around finding optimal locations to insert comments. This could be approached by investigating the comment distribution of danmu videos since this is expected to reveal areas of likely user interest on the video timeline which could provide pointers for preferred locations for the automated creation of danmu comments.

Model	p	R@1	R@5	R@10	MR	MRR	P@5	P@10
Livebot	0 %	6.56 ± 0.05	22.23 ± 0.22	31.36 ± 0.29	22.15 ± 0.37	16.6 ± 0.48	6.44 ± 0.18	6.58 ± 0.18
Livebot-t	0 %	7.09 ± 0.06	24.78 ± 0.23	37.77 ± 0.36	19.86 ± 0.46	20.4 ± 0.48	6.89 ± 0.18	8.02 ± 0.18
MCVT-zero	0 %	**8.79** ± 0.07	27.25 ± 0.25	45.58 ± 0.44	**18.28** ± 0.33	25.6 ± 0.51	8.45 ± 0.18	**8.85** ± 0.20
MCVT	0 %	8.65 ± 0.07	**27.36** ± 0.25	**47.90** ± 0.44	18.81 ± 0.33	**25.8** ± 0.52	**8.70** ± 0.19	8.68 ± 0.19
Livebot	5 %	6.49 ± 0.05	23.49 ± 0.22	32.88 ± 0.31	21.59 ± 0.34	17.4 ± 0.48	6.15 ± 0.19	6.74 ± 0.18
Livebot-t	5 %	9.13 ± 0.08	25.34 ± 0.23	39.40 ± 0.38	19.51 ± 0.34	25.7 ± 0.48	8.90 ± 0.21	8.59 ± 0.21
MCVT	5 %	**19.74** ± 0.18	**42.44** ± 0.4	**56.70** ± 0.55	**12.90** ± 0.35	**32.1** ± 0.64	**18.75** ± 0.36	**19.11** ± 0.38
Livebot	30 %	13.11 ± 0.13	28.45 ± 0.27	41.50 ± 0.40	19.93 ± 0.37	26.0 ± 0.47	12.88 ± 0.24	11.59 ± 0.24
Livebot-t	30 %	13.75 ± 0.13	28.19 ± 0.27	45.59 ± 0.44	18.71 ± 0.35	27.5 ± 0.48	13.14 ± 0.27	13.07 ± 0.27
MCVT	30 %	**24.36** ± 0.22	**47.77** ± 0.46	**61.38** ± 0.59	**11.87** ± 0.31	**36.4** ± 0.59	**24.85** ± 0.41	**24.15** ± 0.42
Livebot	50 %	13.27 ± 0.12	27.17 ± 0.26	41.98 ± 0.40	20.44 ± 0.37	27.8 ± 0.44	13.37 ± 0.29	13.09 ± 0.27
Livebot-t	50 %	13.31 ± 0.12	29.74 ± 0.29	47.07 ± 0.46	18.39 ± 0.34	29.1 ± 0.51	15.59 ± 0.31	16.23 ± 0.32
MCVT	50 %	**26.75** ± 0.25	**48.23** ± 0.46	**62.57** ± 0.60	**11.23** ± 0.29	**37.8** ± 0.67	**26.17** ± 0.42	**26.89** ± 0.42
Livebot	70 %	14.35 ± 0.14	27.59 ± 0.26	42.09 ± 0.41	19.13 ± 0.36	28.1 ± 0.48	15.15 ± 0.34	14.76 ± 0.34
Livebot-t	70 %	15.85 ± 0.14	32.22 ± 0.31	55.44 ± 0.53	18.11 ± 0.36	29.5 ± 0.48	16.77 ± 0.35	17.01 ± 0.35
MCVT	70 %	**27.38** ± 0.25	**51.04** ± 0.49	**63.21** ± 0.61	**11.10** ± 0.27	**39.1** ± 0.71	**28.25** ± 0.43	**27.65** ± 0.42
Livebot	100 %	18.83 ± 0.16	34.50 ± 0.33	52.17 ± 0.51	17.81 ± 0.36	34.7 ± 0.48	18.88 ± 0.36	18.31 ± 0.36
Livebot-t	100 %	17.17 ± 0.15	32.89 ± 0.31	52.91 ± 0.51	18.09 ± 0.36	33.2 ± 0.48	18.15 ± 0.36	18.11 ± 0.36
MCVT	100 %	**29.65** ± 0.28	**55.36** ± 0.53	**63.90**± 0.62	**10.81** ± 0.29	**40.8** ± 0.65	**29.79** ± 0.43	**29.82** ± 0.43

Table 5.2 Results of the comment generation module, model performance is measured using metrics of R@k, P@k, MRR (higher is better, showed in percentage) and MR (lower is better), *p* is the percentage of the preserved comments applied in test set.

Model	p	R@5	MR
Livebot	0 %	22.23 ± 0.22	22.15 ± 0.37
Livebot-t	0 %	24.78 ± 0.23	19.86 ± 0.46
MCVT	0 %	**27.36** ± 0.25	**18.81** ± 0.33
Livebot	30 %	28.45 ± 0.27	19.93 ± 0.37
Livebot-t	30 %	28.19 ± 0.27	18.71 ± 0.35
MCVT	30 %	**47.77** ± 0.46	**11.87** ± 0.31
Livebot	100 %	34.50 ± 0.33	17.81 ± 0.36
Livebot-t	100 %	32.89 ± 0.31	18.09 ± 0.36
MCVT	100 %	**55.36** ± 0.53	**10.81** ± 0.29

Table 5.3 Results of the comment generation module, model performance is measured using metrics of R@k, P@k, MRR (higher is better, showed in percentage) and MR (lower is better), *p* is the percentage of the preserved comments applied in test set.


大哈说二哈
小哥和他的朋友都笑了，
于是尝试一下把饼干递到土拨鼠的嘴巴上
the dodo
当我第一次在马厩发现他的时候 还以为它是只小兔子
When I found Tiny Tim in the barn, I thought it was actually a rabbit.
我一开始也难以相信，因为我觉得...

	P	Model Output
Subtitle: The guy and his friend laughed and pass the biscuit to the groundhog's mouth.	0%	吃土拨鼠23333 Eating groundhog lol.
	5%	我看老鼠要笑到缺氧 I think the mouse is going to laugh until hypoxia.
	30%	我看老鼠要笑到缺氧 I think the mouse is going to laugh until hypoxia.
	100%	被土拨鼠洗脑了哈哈哈哈哈 Brainwashed by groundhog hahahahaha.
When I found him in the barn, I thought it was a rabbit.	0%	你以为他是只兔子啊 Do you believe he is a rabbit?
	5%	这个腿也太长了 The legs are too long.
	30%	小毛驴长得好好看 The little donkey looks very good.
	100%	养出感情的动物就像家人一样 Animals you raised are like family.
I found it hard to believe at first, because I think...	0%	我忍你很久了 I have endured you for a long time.
	5%	明明是你在笑 Obviously you are laughing.
	30%	我要被笑死了哈哈哈哈哈 I'm going to be laughed to death hahahaha.
	100%	主持人的声音更好笑，怎么忍住不笑的 The host's voice is even funnier, how can they resist it?

Fig. 5.4 Several examples of the test set, left side is the video frame and the subtitle translation of the time point. The table on the right shows the generated comment for different preserve rates p.

Chapter 6

Danmu Video Highlight Prediction

In this chapter, we investigate the distribution pattern of danmu comments in videos and attempt to address the second research question ([RQ2] Can we automatically identify appropriate locations in video timelines to insert comments?). Although research progress has been made around analysing danmu comments (Lv et al., 2019a, Ma et al., 2019, Chaoqun et al., 2020, Wang et al., 2020b), understanding user motivation underlying the addition of new danmu comments is little studied. Comments from individual viewers can be highly subjective and unpredictable, but their collective distribution over the video timeline exhibits interesting patterns. For example, these patterns can include brief moments of high danmu density and longer periods of low danmu density (see Fig. 6.1). To some extent, this distribution represents the viewer's interest in different parts of the video. Of particular interest are the high danmu density moments, also known as "Danmu Bursts" in (He et al., 2017). Even though there is no explicit way to define those danmu bursts in the current literature, these densely commented moments are worth studying in detail as they indicate high user engagement. In general, understanding patterns of user danmu contributions based on video content has the potential to be used to create teasers of content in individual videos, which can be used in recommender systems or provide pointers to desired locations for automated danmu comment creation to encourage engagement with content which is likely to be of interest to viewers.

We hypothesize that the pattern of danmu distribution can be predicted based on extracted features from the video. To investigate this hypothesis, we propose a novel **danmu density prediction task**: given a video with zero, little or many comments, can we predict the comment distribution in the future?

The state of the art around this application is very thin, as only one previous investigation attempted to address this similar problem (Zheng et al., 2020). In this chapter, we show that the architecture from the Danmu Generation task 5.2 that leverages surrounding video comments, subtitles, visual and audio signals in a transformer module can be re-purposed to predict the future damnu comment density at any point in the video. Extensive experiments on the expanded danmu video dataset (see 3.3) show that our system can produce accurate predictions, even in the cold case scenario, where only the video content is available.

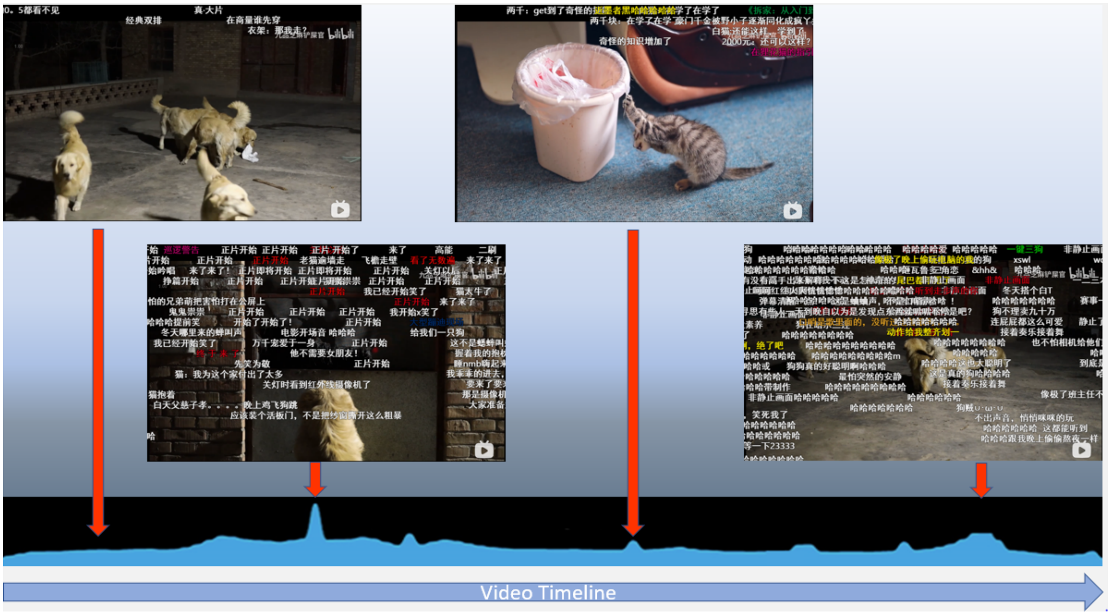


Fig. 6.1 An illustration of a danmu video with scenes of different comment densities. The blue plot on bottom indicates how danmu comments are distributed over the video timeline. Video screenshots on the top are chosen from four different video display timestamp.

Note that an earlier version of this work was published at CBMI 2021 (Wu et al., 2021d). We revised several details of the system in ICMI 2021 (Wu et al., 2021a) and present here the corrected version.

6.1 Related Work

In Chapter 2, we covered literature on social media highlight detection where the general goal is to detect the most interesting and important clips of a video. This goal is similar to our objective of finding the most appropriate locations for automated commenting. In those methods, social video segments are first analyzed and processed into visual features, and then mapped to the target highlight type. In particular, Godi et al. (2017) propose to use pre-trained CNN visual features and LSTMs to detect highlights from ice-hockey videos.

Similar to our method of leveraging video meta-data to estimate the groundtruth of danmu video highlights, self-supervised approaches have been widely applied to address social video highlight detection. (Xiong et al., 2019) propose to use video duration as a weak training signal for detecting highlights of YouTube and Instagram videos. The authors observe that individual clips from a short video are likely to be highlights of that video, whereas clips from long videos are unlikely to be highlights. This can then be used to obtain groundtruth labels when comparing the highlight probability between two random clips. Yang et al. (2015) propose a semi-supervised approach for detecting highlight clips from edited web videos. The key idea is that the highlighted sub-clips within

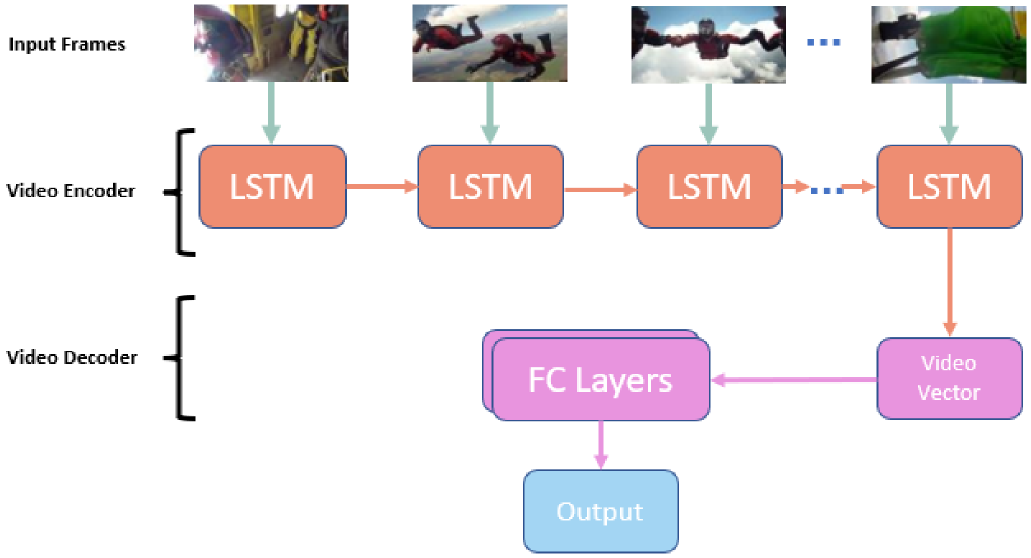


Fig. 6.2 The danmu video popularity detection module reproduced from (Zheng et al., 2020).

a video are commonly present among edited videos while less interesting ones appear less frequently. The system learns what a highlight based on commonalities is among videos in the dataset. They use pre-trained 3D-CNN features as frame features and design an LSTM-based auto-encoder to distinguish highlights. The scenes that commonly appear in the dataset are more likely to be reconstructed while others are not. In this way, they can distinguish highlights by looking at the reconstruction error.

We also note that multi-modal information is not fully explored in social video highlight detection (Xiong et al., 2019, Yang et al., 2015, Godi et al., 2017). They only include visual signals as model input in their corresponding tasks and ignore other modalities such as audio soundtrack or textual video meta-data.

Like social video highlights, in danmu videos, we can clearly observe the phenomenon of danmu bursts (shown in Figure 6.1). These danmu bursts are associated with important moments in a video, thus becoming a potential video highlight measure. Moreover, the distribution of danmu comments can provide fairly accurate signals of user attention, which allows us to predict overall danmu comments distribution. This task of detecting danmu video highlights is largely unexplored, the only available work relates to Zheng et al. (2020). Their objective is to predict which segment in a newly generated danmu video stream among the audiences will be popular. To solve this, they use the danmu density in a video segment to indicate the level of video popularity, since they assume these two metrics are positively correlated. Specifically, they represent the popularity of any 15 seconds video clip with the number of danmu comments, capped to 15, and then normalized this to have a popularity indicator between 0 and 1. The network architecture of Zheng et al. (2020) is illustrated in Figure 6.2 and follows roughly the same architecture as in (Yang et al., 2015, Godi et al., 2017). The danmu videos are first divided into several segments. Keyframes are extracted from the video segments and an

LSTM based architecture is used to summarise visual signals. For predicting the danmu density, they use fully connected layers to transfer the video vectors into a fixed dimension.

6.1.1 Remarks

One issue relating to the approach of Zheng et al. (2020) is that they try to predict the actual number of comments published in a video segment independently of the rest of the video. This is tantamount to predicting whether a clip will be the most popular segment in the video without analysing the rest of the video. This issue could potentially be addressed by feeding larger video segments to the model. However, this would also significantly increase the model size, and this approach is therefore unsatisfactory.

Another issue with their definition of video danmu density arises in the normalisation step. Even though their method is useful in managing data imbalance, it is limited in some aspects. First, when the video is densely commented, most video segments will have more than 15 comments. Also, due to the lagging effect(the phenomenon that user comments usually lag behind their correspond scenes in video displaying timelines (He et al., 2017)) that commonly exists in danmu videos, the danmu density of a precise video segment does not always reflect the degree of popularity.

We notice that all the literature mentioned above shares a common network architecture, which involves an LSTM-based encoder that summarises video frames into a video vector, and a decoder consists of fully-connected layers. In our proposed system, we adopt a similar encoder-decoder structure, but replace the LSTM modules with a Transformer architecture as it is effective in both processing time-series data (Neimark et al., 2021) and fusing multi-modal signals (Tan and Bansal, 2019).

6.2 Proposed System

6.2.1 Damnu Density Definition

In order to address the issues with the normalization method in the previous work (Zheng et al., 2020), we propose to re-define danmu density in the form of a histogram. We define the danmu density d_i for a clip $[i-1,i]$ as the proportion of danmu comments over the duration L of the video:

$$d_i = \frac{n_i}{\sum_{j=1}^{L} n_j}, \tag{6.1}$$

where n_i is the number of danmu comments published between video displaying time $i-1$ and time i. We consider here video clips of 1-second duration as in the previous chapter. The value of d_i indicates the proportion of danmu comments in this second. As danmu comments are stochastic in nature and that some videos have only a few comments, we smooth the histogram by applying a moving average over a 5s window.

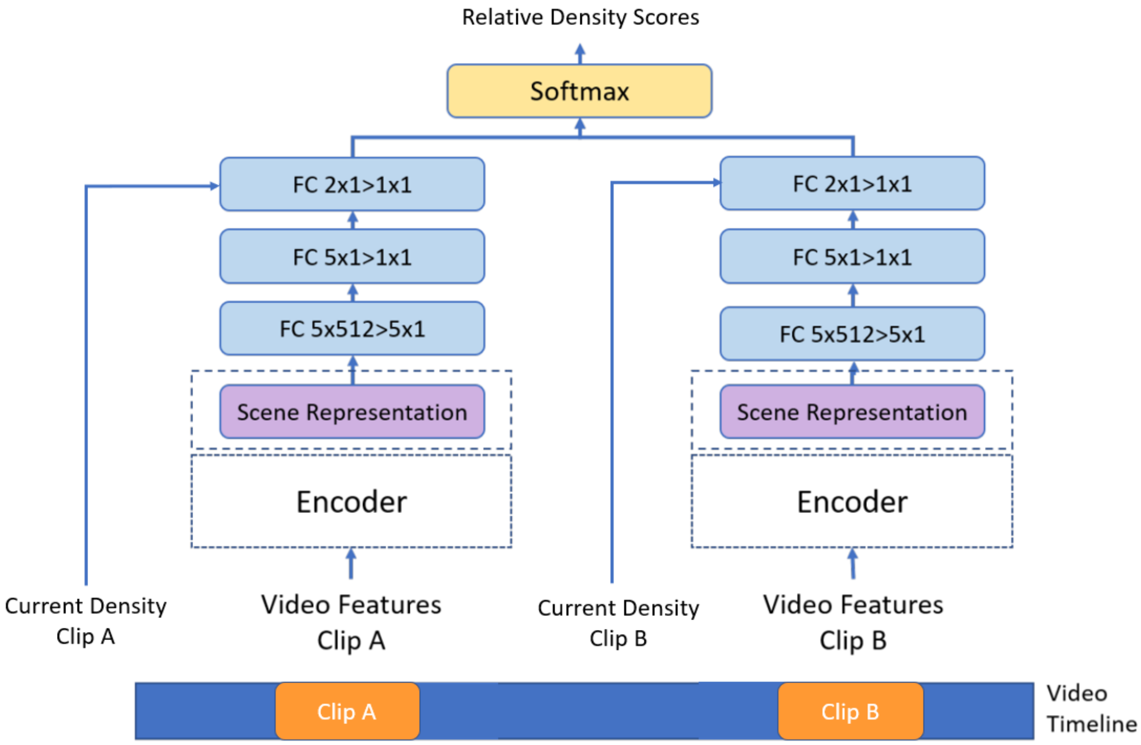


Fig. 6.3 Architecture of our proposed model. The system takes two video clips at the same time, they are encoded respectively (encoder follows the same structure as in Section 5.2) and decoded through three dense layers. Finally, the model output is combined with the current density score and a softmax layer is used to compute the relative density scores.

The goal of the *density prediction task* is now to predict the future comment density from the video content and the current state of existing comments. To retrieve videos with different comment densities, we use the same strategy introduced in our previous chapter(see Section 5.1.1), by removing some of the comments and only keeping a subset of comments $\mathbf{C}_p$ (recall that p denotes the percentage of preserved comments). The "future" comment distribution can be seen as danmu comment densities before removal: $D = \{d_0, \dots, d_L\}$.

Our proposed task is thus defined as follows. For a video $\mathbf{V} = \{\mathbf{s_0}, \dots, \mathbf{s_L}\}$ and a time point i, can we predict the danmu density d_i (before removal), with the k previous seconds of the video clip $\mathbf{s_{[i-k,i]}}$ and the preserved comment set $\mathbf{C}_p$?

6.2.2 Proposed Neural Network Architecture

Since previous work in social video highlight detection (Xiong et al., 2019, Godi et al., 2017) and danmu density prediction (Zheng et al., 2020) is based on LSTM architectures, we propose here to use our Transformer architecture of Section 5.2 to address this problem. The idea is that comment generation and density prediction tasks are related in some ways (see next chapter), thus we choose to re-purpose the transformer-based video encoder in this task.

Our architecture is shown in Figure 6.3. The video features (visual frames, audio soundtracks, surrounding comments and video subtitles) are first encoded into a scene representation through an encoder, the encoder follows roughly the same structure of our previous work (see Section 5.2). Then, two fully-connected layers are used to decode the encoder output into a one-dimensional output. Before computing the final output, we also include the current density score based on the preserved comment set $\mathbf{C}_p$ into the prediction of the *logit*. To do this, we linearly combine the current density value with the decoder output.

In contrast to the previous work (Zheng et al., 2020), where they aim to predict danmu density by analysing individual video segments, a key insight of our method is to realise that we are actually attempting to predict a probability of distribution rather than a value of density. For that reason the analysis of the rest of the video is necessary. We propose to approach the density prediction problem both temporally and globally, by adopting a ranking strategy: each time we analyze two video segments that are randomly selected from the same video, and compute a relative density score of these two segments as a model output. In this way, the system is no longer bounded by a limited video segment and can, to some extent, perceive global information along the video timeline.

For instance, consider clips A (spanning the video interval $[a-k,a]$) and B (spanning the video interval $[b-k,b]$) with their associated groundtruth density scores $d_A = n_a / \sum_j n_j$ and $d_B = n_b / \sum_j n_j$. The predicted density pair, after softmax, corresponds to relative probabilities and can be compared against the relative groundtruth densities:

$$\hat{d_A} = d_A / (d_A + d_B) \tag{6.2}$$

$$\hat{d_B} = d_B / (d_A + d_B) \tag{6.3}$$

A natural choice for the loss function $\mathscr{L}$ between these predicted relative probabilities and the actual groundtruth relative densities is to take the KL divergence, since it provides an explicit measurement of the differences between two probability distributions.

Note that we could potentially consider using more than two clips at a time, but this would considerably increase the memory load when training the model. Also, we could alternatively choose to predict the binary comparison $d_A < d_B$ and use the binary cross-entropy as a loss function. However, since we have access to the actual values of d_A and d_B we might as well use the KL divergence, since it is better suited when $d_A \approx d_B$.

6.2.3 Training Details

For the textual inputs including surrounding comments and subtitles, we set the vocabulary size to 50,000 and set the max length of the input text sequence to 80. In the model, the text embedding is of size 512 and is randomly initialized before training. The dimension of the audio's GRU hidden state is set to 512. For each transformer component, the hidden state dimension is set to 512, the feed-forward network dimension to 2048, the number of heads to 8 and the number of blocks to 6.

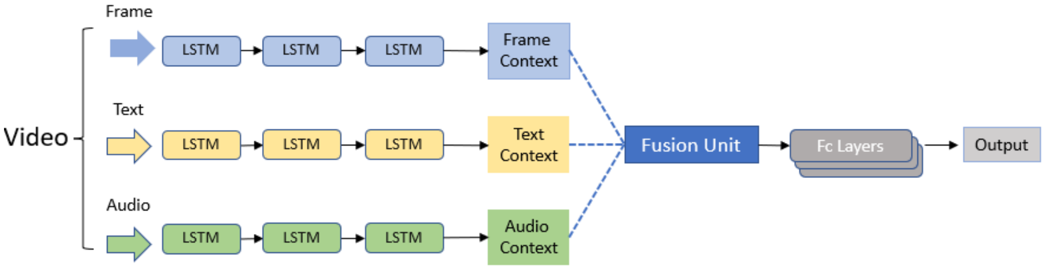


Fig. 6.4 Structure of the baseline model. The encoder structure is sourced from MA-LSTM (Xu et al., 2017b). We modify the original video captioning decoder with fully-connected layers.

The loss criterion we use is KL-Divergence. During training we set the number of epochs to 20, the batch size to 64 and we use the Adam optimizer (Kingma and Ba, 2014) with settings $\beta_1 = 0.9$, $\beta_2 = 0.998$, weight decay $=1 \times 10^{-4}, \varepsilon = 1 \times 10^{-8}$ and learning rate 1×10^{-4}.

The training was done on a Linux server with a single RTX 2080 Ti graphic card, 16 cores Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz and 256GB RAM. The neural network was implemented using Pytorch 1.4.0 and Python 3.6. With the above settings, it took around 24.5 hours to complete a training process of 20 epochs.

6.3 Experimental Investigation

We conduct substantial experiments on the expanded danmu video dataset 3.3. In this section, we first define the scope of our investigation, and then introduce the evaluation metrics and present our experimental results.

6.3.1 Scope of Experimental Study

The experiments are designed to evaluate our proposed system in four aspects:

- The advantages of our proposed model over an LSTM baseline architecture (see Section 6.3.2).
- The impact of the size of the video analysis window on the model performance.
- The impact of the preserve rate p on the model performance.
- The contribution from each of the modalities.

These four strands are carefully investigated in the experimental investigations described in the following sections.

6.3.2 Baseline LSTM Architecture

As both of our objective and task definition are different from the previous work of (Zheng et al., 2020), a direct comparison is not possible. Considering that their method uses the LSTM architecture as the video encoder, we therefore compare our proposed transformer encoder with an LSTM baseline method to evaluate the effectiveness of our architecture. We choose MA-LSTM (Multi-modal Attention Long-Short Term Memory) (Xu et al., 2017b) as our LSTM baseline. Since this network was originally designed for video captioning, we use only the encoder of the MA-LSTM where three LSTM modules are applied to encode frames, audio and subtitle respectively. A fusion unit is then used to combine features from all modalities. Here we apply a *tanh* non-linearity on the linear transformation of all encoded modalities, obtaining a single decoder initialization. The detailed transformations are as follows:

$$h = tanh(W_c[h_1; h_2; ...h_n])$$

where W_c is the parameters of linear transformation operation, h_n means the hidden state from the n-th modality. For the decoder, we use fully connected layers as in our architecture, the overall structure of the modified MA-LSTM is shown in Figure 6.4.

6.3.3 Danmu Density Prediction Evaluation Metrics

We use the actual distribution of danmu comments as groundtruth during the process of evaluation. To measure how close our model prediction is to the groundtruth, we propose first to use the root mean squared error (RMSE), as it is a standard regression quality metric. This measure has the advantage of measuring the correctness of the prediction of danmu density at each time point.

To measure the ability to capture the overall comment distribution of the video, we report the entropy scores (Kullback–Leibler divergence) of the system. Recall that in this case, we normalize the comment density into a probability distribution through the softmax function in order to compute the entropy score.

6.3.4 Danmu Burst Detection Evaluation Metrics

In order to evaluate the ability to identify heavily commented time points in the danmu video (*e.g.* the second selected time point in Figure 6.1), we first aim to systematically define highlights(danmu bursts) using danmu density scores.

We observe that sometimes the boundary that classifies danmu bursts is unclear as videos exhibit very different profiles (see Figure 6.5). The definition of danmu bursts can be somewhat subjective and differs from video to video. This makes it challenging for us to come up with a universal definition of danmu bursts.

Generally, the danmu density scores (the proportion of the number of comments in 1s video segments) are smaller in long videos (*e.g.* the right example) since there are more video segments, hence setting an arbitrary density threshold is not realistic. We also note that some danmu videos have the danmu comments evenly distributed without concentrating on certain displaying timestamps. One example of this type of video is presented on the right side of Figure 6.5. Even though there seem to be many high spikes in the figure, the actual danmu densities of these spikes are only slightly higher

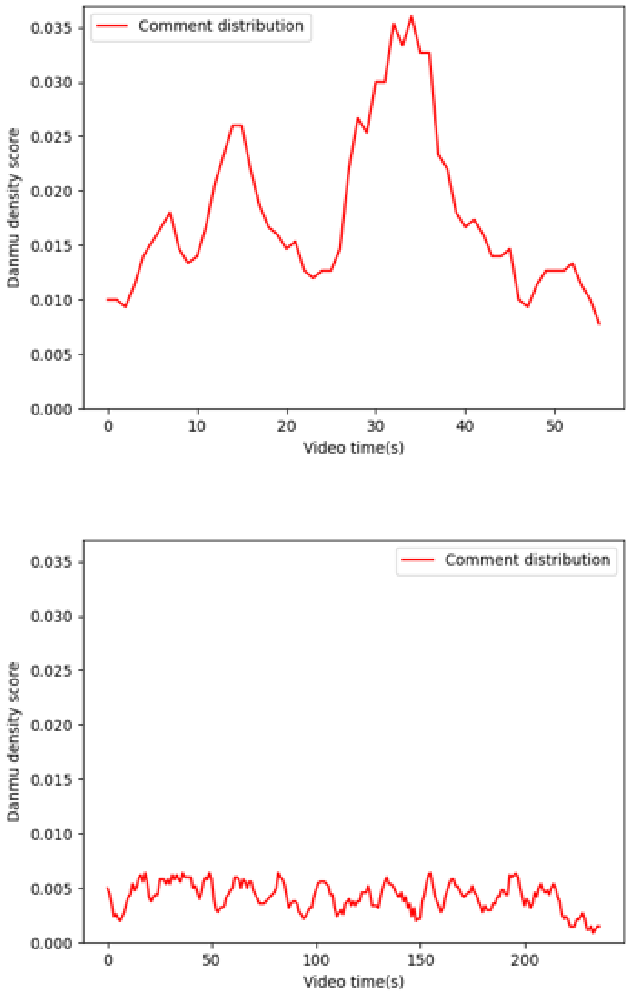


Fig. 6.5 The danmu density scores in each second of the video, each graph corresponds to one type of danmu video.

than average. For example, the highest spike in the lower right figure corresponds to a density score of 0.006. However, the average density score of a 250s video is 0.004 already. Based on the above observations, we finally determine the definition of danmu bursts as follows:

$$d_t = \max(2/L, Quantile(D, 10/L))$$

where d_t controls the density threshold, L denotes the length of the video in seconds, and D means the collection of all density scores in the video. This method is designed to ensure that a burst has to be at least twice that of average damnu density ($2/L$), and also place among the top-10 highest scores.

However arbitrary this definition is, it allows us to introduce retrieval based metrics such as mAP, R@k (recall@k, the percentage of test samples where a heavily commented time point is found within the top-K model output), P@k (precision@k, the percentage of heavily commented time points found in the top-K model output) and NDCG. In this process each test sample is related to a video, and the model is asked to rank all time points in the video by the predicted density scores.

6.3.5 Ablation Study

We evaluate both models for different combinations of modalities, as well as for different window analysis durations. We train the network for three different window analysis sizes $k = 3$, 5 and 7 seconds, respectively. For instance, $k = 5$ means that the model is trained to predict a danmu comment density at a particular time point, based on the previous 5 seconds of video content. Then, we examine the contribution of each modality. We have covered all seven possible combinations of modalities for the input source during testing. For instance, in the case of only the visual modality, the visual input remains unchanged while the audio and text input vectors are filled with zeros. In the experiments, we set the preserve rate p to 0%, 1% and 5%. We do not present here the result for a higher value of p since the comment distribution at a high preserve rate is very close to the groundtruth already.

Table 6.1 presents the performance of both models given the different input conditions. Overall, the results across all seven evaluation metrics consistently indicate that our proposed model exhibits better performance than the baseline method in each evaluation scenario. In particular, when all modalities are available, our relative improvement in mAP over MA-LSTM is 12.63%, 7.06% and 10.29% in complete cold start scenario ($p = 0\%$) and near cold start scenarios ($p = 1\%$ and $p = 5\%$) respectively.

When we gradually remove modalities from the ablation experiments' input, we notice that the most helpful modality is actually the audio soundtrack. In the single modality scenario, audio significantly outperforms the other two modalities. In the double-modalities experiments, we also observe that the performance drops dramatically (6% in mAP) when taking audio away from the input. By contract, the performance loss is trivial when removing text or visual signals from the input source.

To estimate the optimal length of the video analysis window, we test three different values of k in the experiments. We present the model performance under different analysis window lengths k in Table 6.2. In particular, we set k to 3, 5 and 7 seconds. Applying a larger window (*e.g.* $k = 10$) will significantly increase memory thus is not considered in our investigation. There is no clear difference in performance for the different window analysis sizes $k = 5, 7$, but we observe a performance drop (3% in mAP) for shorter input video clips ($k = 3$).

6.3.6 Case Study

To further examine the performance of our proposed model, we show in Fig 6.6 a few examples of our danmu distribution predictions, which are randomly selected from the test set. Each row corresponds to a single video. The first column corresponds to the complete cold start problem ($p = 0\%$); the second column to $p = 1\%$ and the last column to the $p = 5\%$. The groundtruth density (computed at

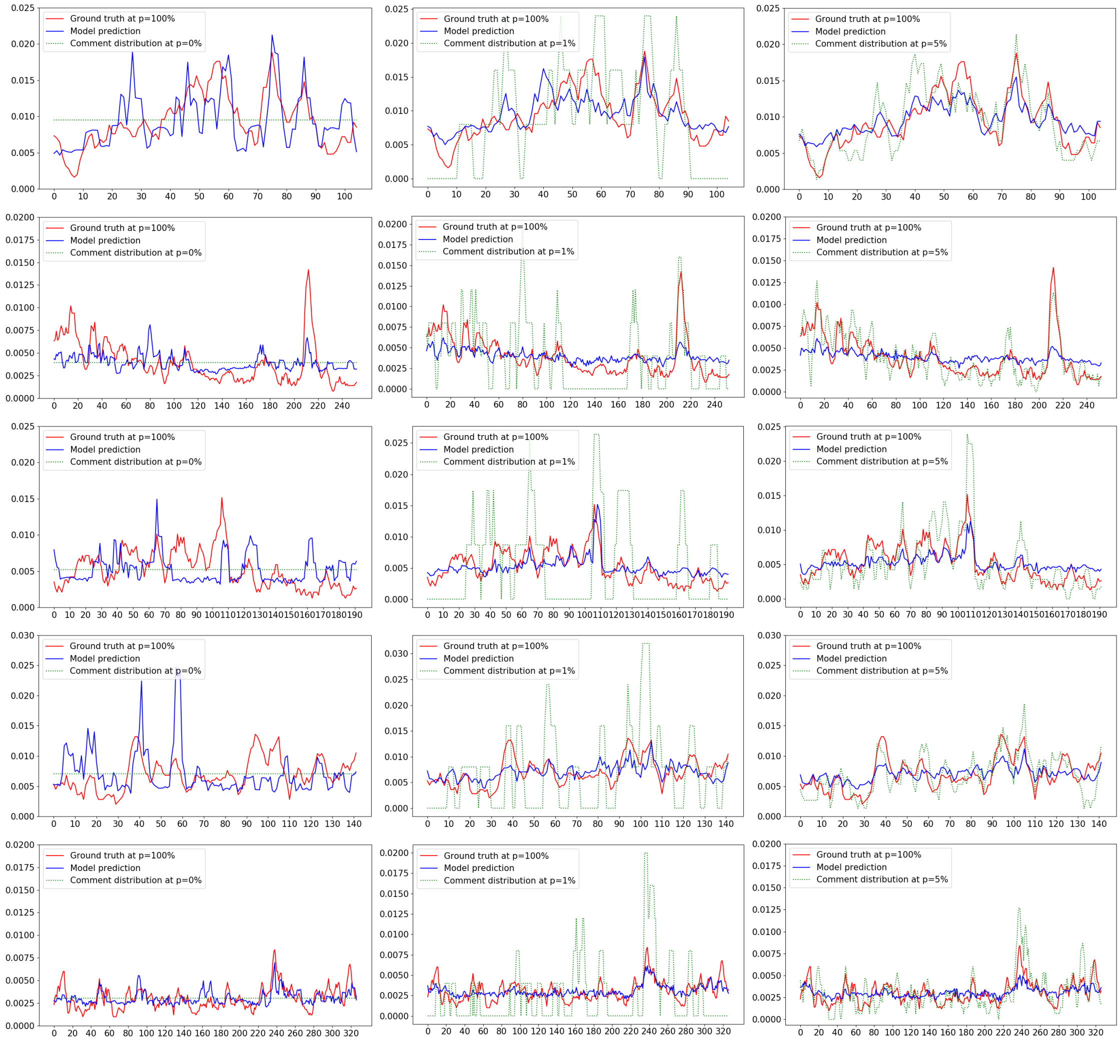


Fig. 6.6 Model predictions vs. ground truth comment distributions of videos in the test set. Model prediction is labeled in blue, groundtruth is labeled in red, the green curve shows the comment distribution at current density setting.

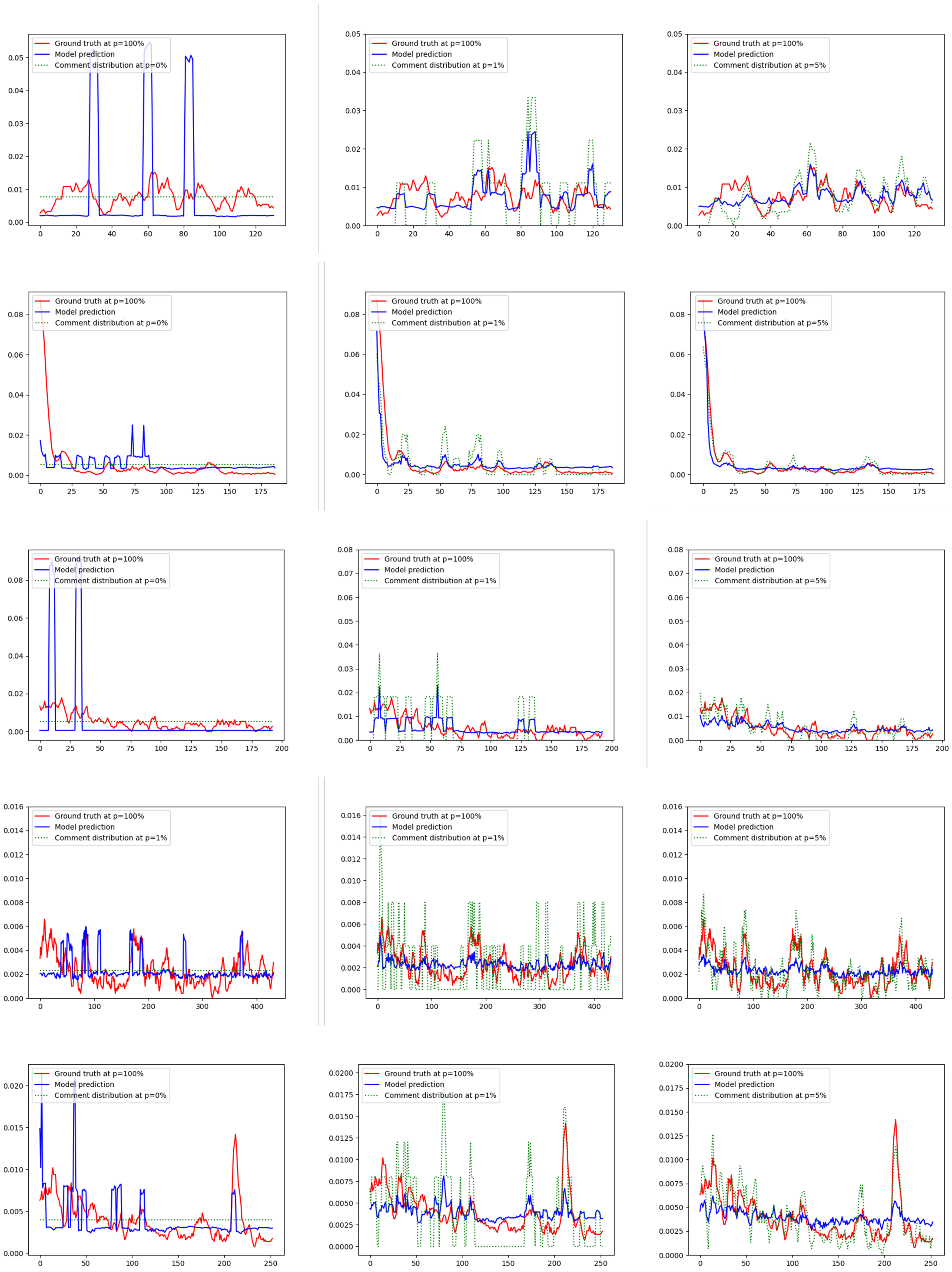


Fig. 6.7 Model predictions vs. ground truth comment distributions of videos in the test set. Model prediction is labeled in blue, groundtruth is labeled in red, the green curve shows the comment distribution at current density setting.

$p = 100\%$) is labelled in red, our prediction in blue, and the current danmu distribution is in dotted green. We can see that our density predictions, even in a cold start scenario, are correlated with the groundtruth distribution.

In the first case (first row of Fig 6.6), our model successfully predicts two danmu burst moments (near $t = 55s$ and $t = 75s$) in the complete cold start scenario, the overall trend of the two lines being clearly correlated. The model prediction in near cold start scenario (*e.g.* $p = 5\%$) is extremely close to the groundtruth distribution. When given a video that is relatively evenly commented (*e.g.* last row in the Figure 6.6), our model still precisely reconstructs the groundtruth.

6.4 Conclusions

In this chapter, we proposed a novel task of reconstructing user danmu comment distributions as a step towards our second research question: Can we automatically identify appropriate locations in video timelines to insert comments? We have proposed to re-use the transformer-based video encoder in this task and compared its performance against an LSTM baseline method.

Our extensive experiments on the expanded danmu video dataset reveal three major conclusions: First, our proposed system constantly outperforms the baseline method, which demonstrates the effectiveness of the Transformer architecture and our multi-modal fusion strategy. Second, we note that in other video highlight detection tasks, visual inputs usually dominates the model performance (Godi et al., 2017, Xiong et al., 2019). However, the results from the ablation investigation indicate that the audio soundtrack is the most contributed modality. This finding supports the idea of including audio soundtrack and subtitles into the system to complement the video representation. Generally speaking, our proposed system produces accurate predictions on classifying danmu bursts. In the complete cold start case, it achieves an mAP score of 57.69% and also presents good approximations of comment distributions during the case study. To this extent, we have partly solved the second research question of finding appropriate locations to insert comments.

In the next chapter, we continue to explore RQ2 and leverage detected danmu video highlights in the automated danmu commenting module to see how the choice of location influences the generated danmu content.

Input	Model	p(%)	RMSE	KLD	R@1	R@5	P@5	mAP	NDCG
Visual only	MA-LSTM	0	3.71e-1	3.91e-1	2.0	3.0	2.6	0.3244	0.7551
	MA-LSTM	1	9.21e-2	1.55e-1	3.5	6.0	3.9	0.3116	0.8045
	MA-LSTM	5	5.73e-2	9.71e-2	4.0	7.5	4.8	0.3791	0.7712
	Transformer	0	2.11e-1	3.71e-1	0.5	2.0	3.2	0.3190	0.7791
	Transformer	1	9.23e-2	9.65e-2	4.0	7.5	5.2	0.3525	0.7905
	Transformer	5	**6.51e-2**	**8.05e-2**	**4.0**	**8.0**	**5.4**	**0.3892**	**0.8133**
Audio only	MA-LSTM	0	9.57e-2	7.71e-2	4.0	10.5	5.2	0.3728	0.7852
	MA-LSTM	1	5.23e-2	7.35e-2	4.0	9.0	5.9	0.5421	0.8440
	MA-LSTM	5	3.32e-2	5.65e-2	8.5	11.5	8.8	0.5733	0.8939
	Transformer	0	1.09e-2	6.71e-2	6.0	11.5	4.1	0.3529	0.8381
	Transformer	1	6.23e-2	5.36e-2	**9.5**	**13.5**	**8.4**	**0.5931**	0.8954
	Transformer	5	**1.51e-2**	**4.05e-2**	9.0	13.0	7.7	0.5899	**0.9219**
Text only	MA-LSTM	0	2.81e-1	1.99e-1	2.5	4.5	2.7	0.2655	0.7041
	MA-LSTM	1	1.21e-1	1.05e-1	2.5	4.0	3.9	0.3984	0.7955
	MA-LSTM	5	9.73e-2	8.90e-2	4.5	7.0	5.8	0.5188	0.8233
	Transformer	0	1.87e-1	1.71e-1	4.0	6.0	3.8	0.4322	0.8177
	Transformer	1	7.23e-2	9.15e-2	4.0	6.0	4.3	0.5126	0.8271
	Transformer	5	**6.51e-2**	**7.05e-2**	**5.5**	**7.0**	**7.4**	**0.5442**	**0.9189**
Visual & Audio	MA-LSTM	0	4.91e-2	2.93e-2	6.0	10.5	4.8	0.4898	0.8676
	MA-LSTM	1	2.74e-2	1.04e-2	7.0	13.5	4.9	0.5957	0.8930
	MA-LSTM	5	1.07e-2	8.71e-3	9.5	14.0	5.8	0.6784	0.9139
	Transformer	0	3.32e-2	1.12e-2	6.0	16.0	6.7	0.5643	0.8932
	Transformer	1	2.58e-2	9.39e-3	8.0	14.0	7.4	0.6894	0.9191
	Transformer	5	**9.97e-3**	**7.65e-3**	**9.5**	**14.0**	**7.9**	**0.6944**	**0.9203**
Visual & Text	MA-LSTM	0	7.34e-2	5.50e-2	3.5	8.0	4.1	0.4536	0.8139
	MA-LSTM	1	4.56e-2	3.28e-2	6.0	10.0	6.8	0.5119	0.8765
	MA-LSTM	5	3.08e-2	1.94e-2	6.5	13.5	6.8	0.5933	0.8998
	Transformer	0	6.99e-2	5.07e-2	6.0	9.5	5.1	0.5143	0.8332
	Transformer	1	5.51e-2	4.31e-2	6.5	13.5	5.7	0.6331	0.8977
	Transformer	5	**3.30e-2**	**3.83e-2**	**8.0**	**14.0**	**7.7**	**0.6480**	**0.9012**
Audio & Text	MA-LSTM	0	4.78e-2	2.71e-2	6.0	11.0	5.1	0.5105	0.8573
	MA-LSTM	1	3.31e-2	9.23e-3	8.0	16.0	7.3	0.5984	0.8892
	MA-LSTM	5	9.01e-3	8.85e-3	9.0	16.0	8.1	0.6797	0.8904
	Transformer	0	5.03e-2	2.21e-2	8.5	13.0	5.8	0.5784	0.8795
	Transformer	1	3.11e-2	1.09e-2	10.0	16.5	9.2	0.6330	0.8928
	Transformer	5	**8.96e-3**	**8.32e-3**	**11.0**	**19.0**	**9.7**	**0.7104**	**0.9253**
Visual & Audio & text	MA-LSTM	0	3.54e-2	1.91e-2	8.0	14.5	9.2	0.5122	0.8975
	MA-LSTM	1	3.17e-2	1.35e-2	8.5	16.0	9.8	0.6510	0.8945
	MA-LSTM	5	8.33e-3	6.67e-3	9.0	18.5	10.4	0.7384	0.9177
	Transformer	0	2.57e-2	8.71e-3	10.0	16.0	7.7	0.5769	0.9077
	Transformer	1	2.32e-2	8.65e-3	10.0	18.5	9.2	0.6970	0.8971
	Transformer	5	**6.61e-3**	**4.05e-3**	**12.5**	**24.0**	**12.4**	**0.8144**	**0.9319**

Table 6.1 Performance of both our proposed model and the baseline approach for different combination of modalities. The accuracy of the danmu comment density prediction is measured in RMSE, KL-Divergence (lower is better), R@k, P@k, mAP, NDCG (higher is better).

k (s)	*p* (%)	RMSE	KLD	R@1	R@5	P@5	mAP	NDCG
3	0	4.23e-2	1.03e-2	6.0	12.0	5.4	0.5334	0.8631
	1	3.03e-2	8.85e-3	7.5	12.5	7.8	0.6299	0.8929
	5	1.01e-3	6.74e-3	9.5	16.5	9.3	0.7883	0.9228
5	0	2.57e-2	8.71e-3	10.0	16.0	7.7	0.5769	0.9077
	1	2.32e-2	8.65e-3	10.0	18.5	9.2	0.6970	0.8971
	5	6.61e-3	4.05e-3	12.5	24.0	12.4	0.8144	0.9319
7	0	2.81e-2	8.89e-3	8.0	16.0	6.9	0.5601	0.8970
	1	2.39e-2	6.70e-3	9.0	16.0	8.3	0.5937	0.9010
	5	7.66e-3	3.89e-3	9.5	22.0	10.8	0.8064	0.9307

Table 6.2 Performance of our proposed model with different video analysis window sizes (length of input video clip in second) and preserve rates.

Chapter 7

A Unified Multi-Task Approach

In previous chapters, we explored the possibilities of both generating comments and predicting danmu video highlights in different stages of cold start scenarios. In this chapter, we continue to explore RQ2: Can we automatically identify appropriate locations in video timelines to insert comments? by considering the actual scenario of commenting at predicted highlight points.

The aim of this chapter is to address the commenting task in its entirety and predict, for any video, both the location and the content of the comments to be published in a single framework. Although the task of highlighting and comment generation can be done separately as we have shown in previous chapters, we propose that both comment generation and highlight detection can be modelled with a single unified framework, built around the same encoder architecture. In this way, given any video analysis window, the system is able to predict both the target comment and the danmu density score of this moment. This approach is not only more technically elegant but also enables communication across two tasks so that the information gained from training one task can potentially help the other.

The proposed system (evaluated in Section 7.3) achieves state-of-the-art performance for both tasks. The results also show that our Multi-Task approach yields better performance than when training the network for each task separately.

The work described in this chapter is published in ICMI 2022 (Wu et al., 2021b).

7.1 Related Work

In recent years, Multi-Task learning has been widely applied in NLP tasks. It leverages the idea that useful information of related tasks can be utilized jointly to achieve simultaneous performance improvement on multiple tasks. The majority of recent Multi-Task models (Wang et al., 2020b, Zhu et al., 2019a, Srivastava et al., 2021) follow a parallel architecture, where the modules for each task run in a parallel way. In order to enable cross-task knowledge sharing, the weights of certain intermediate layers are shared between multiple tasks. In such an architecture, each task has its own task-specific decoding layer, the loss functions of different tasks are generally linearly combined and form a single global loss function thus the entire system can be optimized in conjunction. In this section, we focus on Multi-Task learning applications in Natural Language Generation tasks.

A common objective is to utilize Multi-Task learning strategy to improve the quality (*e.g.* fluency or accuracy) of generated utterances. Zhu et al. (2019a) propose a Multi-Task scheme to tackle natural language generation in task-oriented dialogues. The task aims at generating a dialogue response that contains meaning representation (MR), which refers to key information in the utterance (*e.g.* the location of the nearest gas station in a navigation system). Their proposed system consists of two modules, the first module leverages the high-level guidance by encoding dialogue acts and MR in one GRU module and predicts the target MR. Additionally, in their dataset for each dialogue turn, a human response is also provided, they propose to learn from human responses to improve the fluency of the MR output. This is implemented by training a language model that predicts each token of the human response based on the hidden state of the MR prediction module. In their setting, two modules are trained in a parallel manner with a shared corpus embedding. Their experimental results indicate that adding the language model can largely improve the naturalness of generated utterances. In the field of natural question answering, Srivastava et al. (2021) prove that each related sub-task in the Multi-Task system can benefit from the training of others. They convert a sequence of NLP tasks (such as entity linking detection or question type classification) required for knowledge graph-based question answering (KGQA) into a simultaneous process by adopting a Multi-Task approach. They use a shared BERT encoder to jointly process these upstream tasks and the final prediction module. They show in the experiments that every sub-task is complementary to other tasks and helps the model in performing better towards the final goal of KGQA. Zhou et al. (2019) propose a Multi-Task learning framework in improving the robustness of neural machine translation systems. Their core idea is the knowledge learned from the process of text denoising can be used to supplement the translation procedure. They use one shared transformer encoder that takes noised text as input and has two separate transformer modules that decode the encoded representation into cleaned text and translated text respectively. This system outperforms the vanilla Transformer architecture that is trained on the cleaned text and demonstrates the effectiveness of the proposed Multi-Task approach.

We observe that a common strategy of designing a Multi-Task system is to leverage the auxiliary information of the dataset in improving the performance of the main task. In the danmu video dataset, the video display timestamp is a piece of distinctive side information and has the potential to be utilized in various ways. Wang et al. (2020b) leverage this part of the data in danmu video comment generation. Specifically, they propose VideoIC, which explores the temporal relation between comments and the video content. This integrates the temporal prediction and the danmu comment generation into a Multi-Task framework. Their system aims to predict what the target comment should be given the context window and whether a target comment has been published before or after the input context window. They show that this Multi-Task approach can bring significant improvements to the danmu generation benchmark (improving mean reciprocal rank from 0.291 to 0.312 in their benchmark). When compared to our approach, VideoIC does not look to address the cold video problem, nor does it explicitly look to predict the number of comments in a video segment. In their architecture, the

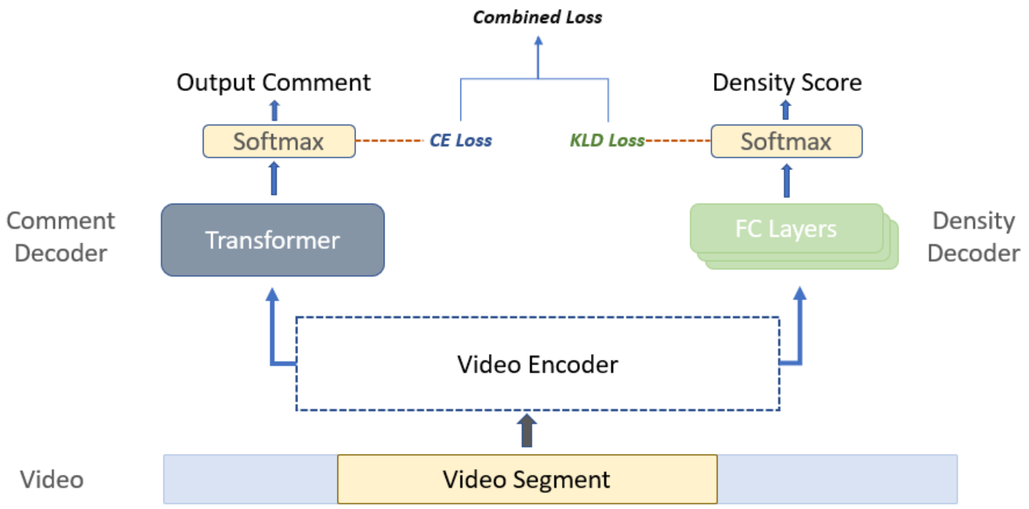


Fig. 7.1 Structure of Multi-Task system, the video encoder is shared for both sub-tasks. The loss functions are combined in the output for system optimization.

temporal prediction sub-task only serves to enhance the video encoding process and the output of this module has no practical purpose. By contrast, in our Multi-Task system, the output of the density prediction module could be further used as the indicator of comment generation module. However, results of VideoIC show that there is something to be gained by approaching the comment content and localisation in a Multi-Task fashion.

Inspired by previous work, we assume that, by applying the Multi-Task learning strategy, the task of danmu density prediction can contribute to the danmu generation module and vice versa.

7.2 Proposed System

7.2.1 Task Definition

The proposed task is defined as follows. For a video, split into segments of 1-second duration and a time point i, can we predict the content of a danmu comment to be inserted at that point, as well as the danmu density d_i, with only knowledge of the previous 5 seconds of the video clip and associated comments (i.e. in the range $[\max(i-5,0),i]$)?

We also consider different danmu density scenarios by following the approach we developed previously in section 5.1.2, and sort the existing comments $\mathbf{C}$ for a video by their publication date, keeping only a subset $\mathbf{C}_p$ consisting of a percentage p of the earliest comments of the video. This allows us to reconstruct video danmu comments at different phases of their lifetime.

7.2.2 Network Architecture

Since both tasks can be trained separately with the same encoder architecture, we hypothesise that this transformer-based video encoder 7.2 can simultaneously be trained for both tasks in a Multi-Task fashion. The overall framework is illustrated in Figure 7.1. The encoder, which is shared between the two tasks, takes its input from three modalities (visual, text and audio) and integrates them into a unified representation. We then apply an independent separate decoder for each of the tasks. The comment generation decoder, shown in Figure 7.3, follows the same workflow as we described in Section 5.2. The output comment is generated through a Transformer layer with 4 multi-head attention modules that attend to the masked target comment $\boldsymbol{y}$, text hidden state H_e, visual hidden state H_{ae} and audio hidden state H_{vae} respectively. Then the probability of output comment $\hat{\boldsymbol{y}}$ is produced with an softmax layer on top of the decoder output. We use cross-entropy as the loss function $\mathscr{L}_G$. In the danmu density prediction sub-task, we use fully-connected layers to decode the encoder output into a one-dimensional output. We use KL-Divergence loss $\mathscr{L}_D$ for this module.

The whole framework is then jointly optimized using an end-to-end method.

7.2.3 Multi-Task Learning

We apply a Multi-Task learning approach for the training of our proposed system. The danmu commenting module and density prediction module are jointly trained with a weighting loss function:

$$\mathscr{L} = \alpha \mathscr{L}_D + (1-\alpha)\mathscr{L}_G \tag{7.1}$$

where $\mathscr{L}_G$ and $\mathscr{L}_D$ refer to comment generation loss and density prediction loss respectively. We use α to control the relative weight of each task during training. The value of α ranges from 0 to 1, a higher value of α increases the priority of training the comment generation task during the system optimization.

7.3 Experimental Investigation

This section gives detail of our experimental study, model training, the result obtained and thesis analysis.

7.3.1 Scope of Experimental Study

In our experiments, we focus on investigating the system performance of the following aspects:

- Comparison between proposed unified framework and Single-Task approaches for both tasks.
- The impact of commenting locations on the model performance.

.

Model	p	R@1	R@5	R@10	MR	MRR	P@5	P@10
ONMT-Livebot	0 %	6.56 ± 0.05	22.23 ± 0.22	31.36 ± 0.29	22.15 ± 0.37	16.6 ± 0.48	6.44 ± 0.18	6.58 ± 0.18
Baseline (MCVT)	0 %	**8.65 ± 0.07**	27.36 ± 0.25	47.90 ± 0.44	18.81 ± 0.33	25.8 ± 0.52	**8.70 ± 0.19**	**8.68 ± 0.19**
Unified	0 %	8.54 ± 0.06	**28.30 ± 0.22**	**48.31 ± 0.45**	**18.13 ± 0.31**	**25.9 ± 0.49**	7.89 ± 0.18	8.22 ± 0.18
ONMT-Livebot	5 %	6.49 ± 0.05	23.49 ± 0.22	32.88 ± 0.31	21.59 ± 0.34	17.4 ± 0.48	6.15 ± 0.19	6.74 ± 0.18
Baseline (MCVT)	5 %	19.74 ± 0.18	42.44 ± 0.4	**56.70 ± 0.55**	12.90 ± 0.35	32.1 ± 0.64	18.75 ± 0.36	19.11 ± 0.38
Unified	5 %	**21.53 ± 0.07**	**47.66 ± 0.22**	52.03 ± 0.35	**12.28 ± 0.25**	**33.7 ± 0.41**	**20.69 ± 0.22**	**21.03 ± 0.23**
ONMT-Livebot	30 %	13.11 ± 0.13	28.45 ± 0.27	41.50 ± 0.40	19.93 ± 0.37	26.0 ± 0.47	12.88 ± 0.24	11.59 ± 0.24
Baseline (MCVT)	30 %	24.36 ± 0.22	47.77 ± 0.46	61.38 ± 0.59	11.87 ± 0.31	36.4 ± 0.59	24.85 ± 0.41	24.15 ± 0.42
Unified	30 %	**26.45 ± 0.08**	**51.23 ± 0.22**	**63.21 ± 0.29**	**10.59 ± 0.27**	**36.9 ± 0.58**	**25.29 ± 0.37**	**25.9 ± 0.51**
ONMT-Livebot	50 %	13.27 ± 0.12	27.17 ± 0.26	41.98 ± 0.40	20.44 ± 0.37	27.8 ± 0.44	13.37 ± 0.29	13.09 ± 0.27
Baseline (MCVT)	50 %	**26.75 ± 0.25**	48.23 ± 0.46	**62.57 ± 0.60**	11.23 ± 0.29	**37.8 ± 0.67**	**26.17 ± 0.42**	**26.89 ± 0.42**
Unified	50 %	24.57 ± 0.06	**50.65 ± 0.41**	61.76 ± 0.29	**11.2 ± 0.2**	37.7 ± 0.91	23.18 ± 0.71	25.19 ± 0.43
ONMT-Livebot	70 %	14.35 ± 0.14	27.59 ± 0.26	42.09 ± 0.41	19.13 ± 0.36	28.1 ± 0.48	15.15 ± 0.34	14.76 ± 0.34
Baseline (MCVT)	70 %	27.38 ± 0.25	51.04 ± 0.49	63.21 ± 0.61	11.10 ± 0.27	39.1 ± 0.71	**28.25 ± 0.43**	**27.65 ± 0.42**
Unified	70 %	**28.79 ± 0.31**	**52.55 ± 0.51**	**64.9 ± 0.64**	**10.31 ± 0.25**	**40.2 ± 0.33**	26.91 ± 0.26	26.46 ± 0.37
ONMT-Livebot	100 %	18.83 ± 0.16	34.50 ± 0.33	52.17 ± 0.51	17.81 ± 0.36	34.7 ± 0.48	18.88 ± 0.36	18.31 ± 0.36
Baseline (MCVT)	100 %	29.65 ± 0.28	**55.36 ± 0.53**	63.90 ± 0.62	10.81 ± 0.29	**40.8 ± 0.65**	**29.79 ± 0.43**	**29.82 ± 0.43**
Unified	100 %	**29.91 ± 0.26**	51.69 ± 0.49	**63.12 ± 0.56**	**10.47 ± 0.47**	39.6 ± 0.51	27.54 ± 0.38	29.21 ± 0.4

Table 7.1 Results table showing the performance for both our proposed framework and the baseline methods on the most commented locations.

To study the performance change when shifting from a Single-Task baseline to a Multi-Tasking approach, we set the Multi-Tasking weighting parameter α to 0 or 1 to get the results of the Single-Task approach for comparison.

To investigate the impact of locations on the danmu generation task, we include two commenting scenarios: (1) creating comments on the most commented locations; (2) creating comments on the predicted highlights.

7.3.2 Training Details

We use the expanded danmu video dataset (see Section 3.3) for the experiments. Similar to the experiments conducted in section 5.1.2 we sample the training data as follows: each target comment for the training set is randomly selected from the original comment set $\mathbf{C}$ and the corresponding comment's context is defined as the 5 nearest comments from the preserved comment set $\mathbf{C}_p$ that precede the target danmu in the video timeline as well as predate its publication date.

The training data is sampled for $p = 0\%, 5\%, 30\%, 50\%, 70\%$ and 100%, to form a combined training set of 4,800,145 pairs of target comment/context comments. Target comments can be sampled multiple times for different contexts.

To evaluate the model performance on its predicted highlights, we first build the test set by taking one test sample from each second of a video. Based on the model predictions of comment density, we select 10 highlighted time points per video specifically for ranking evaluation.

For the text input, we build the vocabulary by selecting the most frequent 50,000 words in the dataset and set the max length of the input text sequence to 50 words. In the model, the text embedding is of size 512 and is randomly initialized before training. The dimension of the audio's

Model	p	R@1	R@5	R@10	MR	MRR	P@5	P@10
ONMT-Livebot	0 %	6.51 ± 0.06	23.21 ± 0.21	34.27 ± 0.31	21.58 ± 0.45	16.1 ± 0.58	6.32 ± 0.38	6.61 ± 0.28
Baseline (MCVT)	0 %	7.42 ± 0.08	29.86 ± 0.28	45.05 ± 0.39	18.09 ± 0.32	26.1 ± 0.48	7.85 ± 0.18	7.65 ± 0.24
Unified	0 %	**8.79 ± 0.12**	**40.3 ± 0.36**	**48.31 ± 0.51**	**16.13 ± 0.65**	**27.8 ± 0.49**	**8.19 ± 0.2**	**8.32 ± 0.15**
ONMT-Livebot	5 %	7.71 ± 0.05	23.59 ± 0.39	34.89 ± 0.31	20.3 ± 0.34	17.9 ± 0.55	7.05 ± 0.19	7.24 ± 0.38
Baseline (MCVT)	5 %	19.83 ± 0.24	39.98 ± 0.44	54.1 ± 0.62	13.21 ± 0.45	30.6 ± 0.72	19.69 ± 0.42	18.88 ± 0.41
Unified	5 %	**24.53 ± 0.57**	**51.36 ± 0.23**	**55.78 ± 0.55**	**11.84 ± 0.34**	**34.7 ± 0.31**	**23.7 ± 0.24**	**23.03 ± 0.55**
ONMT-Livebot	30 %	12.31 ± 0.33	30.35 ± 0.57	41.55 ± 0.51	19.53 ± 0.37	28.8 ± 0.47	11.81 ± 0.22	11.79 ± 0.34
Baseline (MCVT)	30 %	22.59 ± 0.29	48.26 ± 0.76	60.32 ± 0.79	11.91 ± 0.27	36.2 ± 0.29	22.85 ± 0.41	22.15 ± 0.42
Unified	30 %	**25.91 ± 0.17**	**52.33 ± 0.27**	**64.49 ± 0.21**	**10.79 ± 0.22**	**37.1 ± 0.74**	**25.92 ± 0.37**	**25.56 ± 0.71**
ONMT-Livebot	50 %	12.45 ± 0.22	32.37 ± 0.26	42.51 ± 0.42	19.79 ± 0.47	29.5 ± 0.44	12.33 ± 0.39	12.18 ± 0.22
Baseline (MCVT)	50 %	25.49 ± 0.25	**52.56 ± 0.86**	62.55 ± 0.6	**10.79± 0.29**	**37.8 ± 0.67**	**25.17 ± 0.42**	**25.82 ± 0.42**
Unified	50 %	**25.51 ± 0.32**	50.61 ± 0.36	**62.88 ± 0.25**	11.4 ± 0.19	35.4 ± 0.51	25.13 ± 0.72	24.19 ± 0.66
ONMT-Livebot	70 %	15.31 ± 0.24	32.59 ± 0.36	43.02 ± 0.71	18.91 ± 0.46	30.1 ± 0.88	14.65 ± 0.48	14.29 ± 0.49
Baseline (MCVT)	70 %	26.77 ± 0.35	50.65 ± 0.69	62.87 ± 0.71	11.23 ± 0.27	36.9 ± 0.61	26.22 ± 0.43	**26.63 ± 0.42**
Unified	70 %	**27.79 ± 0.31**	**51.55 ± 0.46**	**66.29 ± 0.55**	**10.45 ± 0.25**	**37.9 ± 0.36**	**27.31 ± 0.36**	26.55 ± 0.3
ONMT-Livebot	100 %	17.99 ± 0.36	36.56 ± 0.32	49.73 ± 0.66	17.57 ± 0.64	34.7 ± 0.69	17.8 ± 0.36	17.95 ± 0.54
Baseline (MCVT)	100 %	28.61 ± 0.58	53.2 ± 0.73	64.79 ± 0.62	10.9 ± 0.51	38.8 ± 0.65	28.27 ± 0.43	27.81 ± 0.43
Unified	100 %	**32.23 ± 0.33**	**53.64 ± 0.51**	**67.72 ± 0.58**	**10.31 ± 0.32**	**40.1 ± 0.51**	**31.54 ± 0.32**	**32.19 ± 0.37**

Table 7.2 The result of both our proposed framework and the baseline methods on the predicted highlights from our proposed framework.

GRU hidden state is set to 512. We apply the same setting for all Transformer components used in the network. For each Transformer, the hidden state dimension is set to 512, the position-wise feed-forward network dimension is 2048, the number of heads is 8 and the number of blocks is 6. The loss criterion for comment generation is cross-entropy and KL divergence for density prediction. The relative weight of loss α is set to 0.7, the number of epochs is 8 and the batch size is set to 32. We use the Adam optimizer Kingma and Ba (2014) with settings β_1 = 0.9, β_2 = 0.998, weight decay =$1 \times 10^{-4}, \varepsilon = 1 \times 10^{-5}$ and learning rate 1×10^{-4}. All training was done on a Linux server with a single RTX 2080 Ti graphic card, 16 cores Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60GHz and 256GB RAM. The model is implemented using Pytorch 1.4.0 and Python 3.6. With the above settings, it takes around 41 hours to complete the training.

7.3.3 Evaluation Metrics

Density Prediction Evaluation

Following our previous study in the last chapter, we use the actual distribution of danmu comments as groundtruth for our evaluation. To measure the correctness of the prediction of danmu density at each time point, we use the root mean squared error (RMSE), as it is a standard regression quality metric. We also use the Kullback-Leibler divergence to measure our ability to reconstruct the actual comment distribution. Lastly, we include normalized discounted cumulative gain (NDCG) to measure the ability of ranking highlight time points above others.

Model	p	KLD	RMSE	NDCG
Single Task Baseline	0 %	9.89e-3	2.33e-2	0.895
Unified	0 %	**8.71e-3**	**1.09e-2**	**0.918**
Single Task Baseline	1 %	8.89e-3	2.33e-2	0.895
Unified	1 %	**8.65e-3**	**9.23e-3**	**0.957**
Single Task Baseline	5 %	5.53e-3	6.61e-3	0.930
Unified	5 %	**4.05e-3**	**6.51e-3**	**0.949**
Single Task Baseline	30 %	3.31e-3	6.51e-3	0.961
Unified	30 %	**3.23e-3**	**6.49e-3**	**0.963**

Table 7.3 Danmu density prediction performance for various levels of input comments. Performance between the predicted damnu density histograms and the groundtruth histograms at p=100% is measured as the Kullback-Leibler divergence (KLD), root mean square error (RMSE) and Normalized Discounted Cumulative Gain (NDCG).

Danmu Generation Evaluation

It is widely accepted that reference-based metrics for generation tasks like BLEU and ROUGE are not suitable for the evaluation of video comments (Das et al., 2017, Ma et al., 2019, Zhang et al., 2020). We therefore follow Das et al. (2017) and focus on the ability to rank the correct comment originally appearing at this point in the video over other comments taken from the dataset. We evaluate our system through a retrieval based protocol: the model is asked to re-rank a candidate set for each test sample. The comment set for re-ranking is made of 100 comments, including 5 correct groundtruth comments for this point in the video, the 20 most similar comments to the title of the video based on tf-idf score (plausible candidates), the 20 most frequent comments in the training set and 55 randomly sampled comments. We report the Recall@k, Precision@k, Mean Rank (MR, lower the better) and Mean Reciprocal Rank (MRR) as evaluation metrics on this retrieval task. The confidence interval is reported for each of these metrics with a confidence level at 95% (for R@k, we use the confidence interval for population proportions).

7.4 Results

7.4.1 Danmu Generation Task

Table 7.1 reports the main performance results for our unified architecture(Unified), Livebot implementation (ONMT-Livebot) and a Single-Task baseline that we proposed in Section 5.2(MCVT). We follow here the standard approach of choosing the video timestamps from the most commented locations (at $p = 100\%$) in the video timeline. In both Multi-Task and Single-Task (MCVT) training settings, our architecture offers significantly better results than Livebot. This is principally due to the multi-density training and the leveraging of subtitles and audio. We can also observe that the Multi-Task offers a small but systematic improvement over the Single-Task training. We then looked in Table 7.2 at the more realistic scenario, where the timestamps are sampled according to the *predicted* highlights of our unified network instead of the groundtruth highlights. The same general observations can be made here but we note that the performance for Livebot and MCVT drop slightly when compared to results in Table 7.1. Conversely, the performance of our Multi-Task network increases.

The first observation here is that this correlation between the density prediction and the performance of the danmu generation is a piece of further evidence that both tasks are indeed related. We note that a consequence of this is that our system is better at generating comments at locations where it predicts highlights. In other words, the system seems to be better at talking about a video clip if it is seen as a highlight by our system. Examples of generated comments are presented in Figure 7.4. All three models produce fluent and reasonable output. It can be argued that the output generated from our unified framework is more closely related to the video scene in the cold start cases ($p = 0\%$ and $p = 5\%$).

7.4.2 Density Prediction Results

The results for various values of comment retention p are presented in Table 7.3. Recall that lower KLD and RMSE and higher NDCG correspond to better performance.

We observe that our new architecture compares favourably against our previous density prediction network (see Section 6.2), which is labelled as Single Task Baseline in the result table, in each evaluation scenario.

7.5 Conclusions

In this chapter, we proposed to address the problems of generating live video comments and predicting where to insert them in the video with a single unified Multi-Task framework. Our model is built on the multi-modal fusion Transformer architecture, and exploits the video visual content, subtitles, audio signals, and surrounding comments. In our framework, the two tasks are trained simultaneously with a shared encoder in an end-to-end manner.

The two subsystems were evaluated on a large-scaled danmu commenting dataset against state-of-the-art baseline methods. Our results show that the Multi-Task approach consistently outperforms the Single-Task baselines. This supports our proposal that the information learned from modelling danmu comment distribution can actually benefit the comment generation module and vice versa.

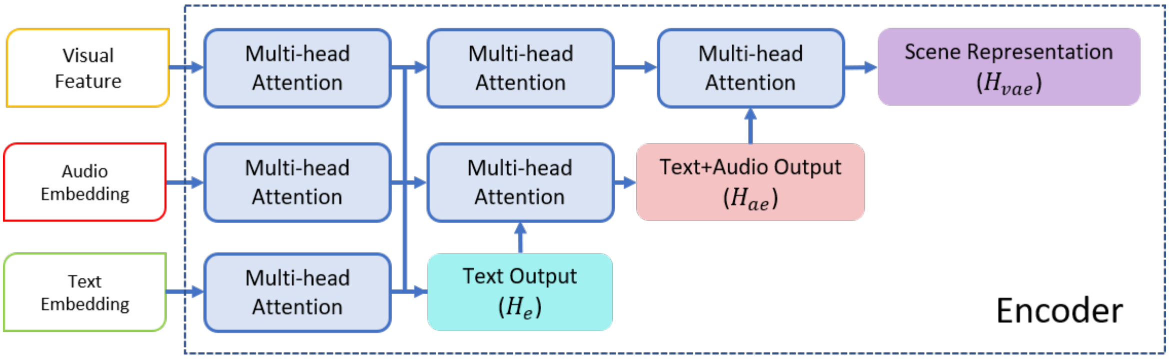


Fig. 7.2 Architecture of the system encoder.

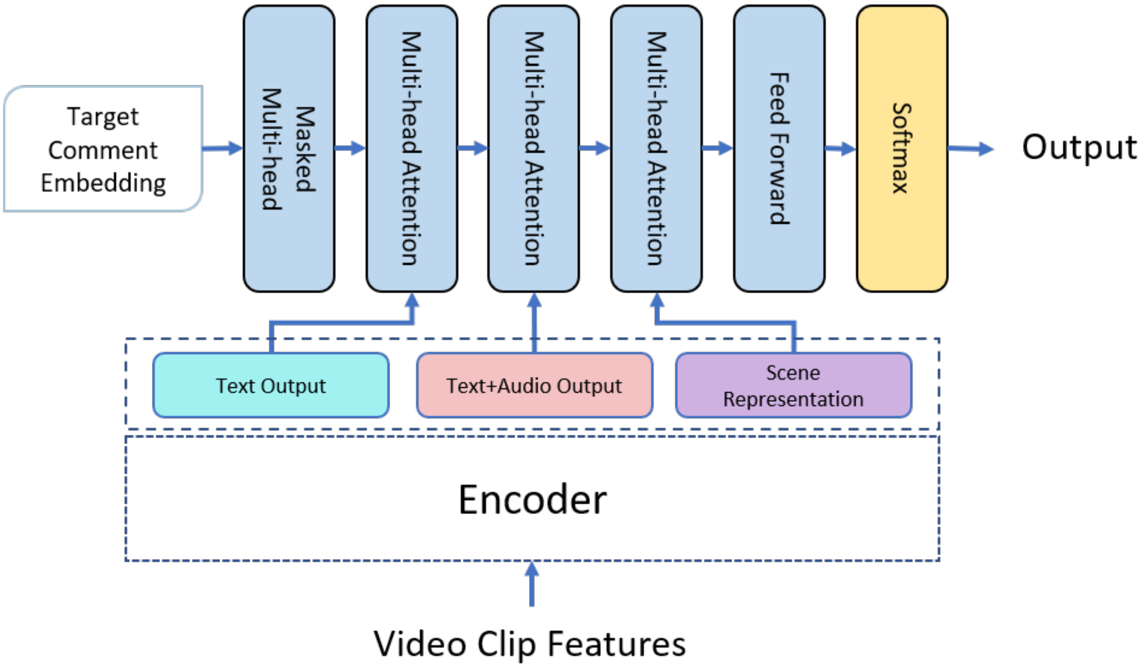


Fig. 7.3 Danmu Generation Decoder Architecture.

真的太可爱了
It is really cute
小哥和他的朋友都笑了，
于是尝试一下把饼干递到土拨鼠的嘴巴上
Subtitle: The guy and his friend laughed and pass the biscuit to the groundhog's mouth.

P	0%	5%	100%
Context	-	小熊猫不是浣熊吗? (Isn't the red panda a raccoon?)	小熊猫不是浣熊吗? (Isn't the red panda a raccoon?) 我特别想握小熊猫的爪子 (I especially want to hold the paw of the red panda) Awsl (That's so cute) 小熊猫站起来了 (The red panda stands up) 看来小熊猫喜欢吃水果 (It seems that red pandas like to eat fruits)
Output	Livebot: 哈哈哈哈哈哈哈 (Hahahahahaha) MCVT: 啊啊啊啊啊啊啊萌! (Ahahahahah cute!) Unified: 怎么可以这么可爱 (How can it be so cute)	Livebot: 小熊猫是吗 (Is it a red panda?) MCVT: 浣熊看起来好可爱 (The raccoon looks so cute) Unified:我的小熊猫只是饿了 (My little panda is just hungry)	Livebot: 小浣熊我抱走了 (I will keep the little raccoon to myself) MCVT: 小熊猫好评哈哈哈哈哈哈 (Red panda praise hahahahahaha) Unified: 小熊猫在这里好受欢迎啊哈哈哈哈 (The red panda is so popular here hahahahaha)

P	0%	5%	100%
Context	-	都是老鼠 待遇差别真大 (They are all rats, the treatment is really different) 这是熊吧? (Is this a bear?)	都是老鼠 待遇差别真大 (They are all rats, the treatment is really different) 这个土拨鼠很漂亮的 (This groundhog is very beautiful) 吃了你们就不给了 (You won't give more it I eat it) 老子，在吃饼干! (I am eating cookies!)
Output	Livebot: 笑死我了 (It is hilarious) MCVT: 吃土拨鼠 2333333 (Eating groundhog lol) Unified: 土拨鼠是也没有吃饱 (The groundhog is not full)	Livebot: 老鼠是老鼠是 (Mouse is mouse is) MCVT: 我看老鼠要笑到缺氧 (I think the mouse is going to laugh until hypoxia) Unified:老鼠不会像这样的 (Mouse won't be like this)	Livebot: 饼干我们这里的好吃 (The biscuits here are delicious) MCVT: 被土拨鼠洗脑了哈哈哈哈哈哈哈 (Brainwashed by the groundhog hahahahahahaha) Unified: 吃饼干的样子好萌好萌 (The way he eats the biscuit is so cute)

P	0%	5%	100%
Context	-	卧槽，爱了，开口跪 (Oh my god, I love it from the first line)	卧槽，爱了，开口跪 (Oh my god, I love it from the first line) 阿桑好怀念 (Sang I missed so much) 好好听啊 (Nice singing) 后面那位年轻小哥长得还可以? (The young man behind is good looking?) 这歌有点年头了啊 (This song is a bit old)
Output	Livebot: 66666 (Good) MCVT: BGM 是什么 (What is the background music?) Unified: 这是我的歌 (This is my song)	Livebot: 哇塞爱了爱了 (Wow, love it love it) MCVT: 这个歌我吹爆 (I would recommend this song) Unified:小姐姐唱的好甜 (This lady's singing is so sweet)	Livebot: 比我唱的好 (This sings better than me) MCVT: 老歌听起来有感觉 (Old songs sound emotional) Unified: 天使吻过的嗓子 (The voice is kissed by an angel)

Fig. 7.4 The generated comments for both our proposed model and baselines, for a video in the test set under different cold start stages.

Chapter 8

System Evaluation

In the last chapter, we showed that our Multi-Tasking solution yields better performance on both the task of comment generation and comment density prediction than solving them individually. However, our Multi-Task system has so far only been evaluated through retrieval-based metrics which concentrate on the ability to rank the plausibility of human-generated comments. The human evaluations proposed in (Ma et al., 2019, Wang et al., 2020b) are not entirely suitable to evaluate our work since these evaluations mainly focus on the fluency and relevancy of the comments. Evaluation of a fully functional annotation system should also incorporate measurement of the appropriateness of the comment locations.

In this chapter, we aim to address our last research question ([RQ3:] How do our automatically created comments compare to human comments for the same videos?) and propose a novel danmu commenting human evaluation with the consideration of our objectives of predicting both content and the insert locations for the comments. The evaluation includes the following criteria:

- *Relevancy*: is the content of the danmu comment relevant to the current context?
- *Timeliness*: is the insertion point of the comment adequate?
- *Informativeness*: does the danmu comment bring new information and knowledge to the current context?
- *Fluency*: is the danmu comment fluent and natural?

Note that Relevancy and Fluency are adopted and introduced in our previous human evaluation experiment in Section 4.3, where we focused on generating human-like comments for videos with various densities. Here, we add another two criteria to extend the evaluation scenario of our unified framework.

Instead of evaluating danmu comments statically, we also implement a real-time evaluation interface to simulate the original user experience of a danmu video website. We demonstrate the use of this evaluation strategy by conducting a human experiment to evaluate the quality of the comments resulting from the system we described in Chapter 7. The results show that the quality of our generated comments is almost indistinguishable from real human danmu comments. Our artificial comments even score higher in Relevancy and Timeliness than human comments.

8.1 Related Work

This section provides an overview of current human evaluation practices in NLG systems, with a special focus on the field of danmu video comments generation.

Almost all previous studies (Ma et al., 2019, Wang et al., 2020b, Chaoqun et al., 2020, Zhang et al., 2020) in danmu commenting have conducted human evaluation sessions to obtain direct assessments of comment quality in multiple desired dimensions. Even though automatic metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and ROUGE (Lin, 2004) are well-established and commonly adopted in NLG methods, these metrics are often criticized for their lack of correlation with human judgements in evaluating conversational content (Liu et al., 2016).

In the work of Ma et al. (2019) and Wang et al. (2020b), the evaluation criteria consist of Fluency, Relevancy and Correctness, the scores are required to be an integer from 1 to 5 (the higher the better). The first two criteria are common choices in the evaluation of most NLG tasks (Sai et al., 2022). Fluency typically refers to the authenticity of the generated text concerning grammar and word choice. Texts with grammar errors, odd collocations or typos will usually receive a low Fluency score. Relevancy is usually used to check how close are the generated text related to the context. The context here refers to both the video content and the danmu comments (if available), comments that are specifically related to the current video scene or engaged with existing contextual comments are favoured. In the danmu commenting task, Correctness measures the confidence that the generated comments have been made by humans, while the interpretation of this metric often varies with the task involved. For example, it typically refers to the response accuracy based on the real-world knowledge in dialogue system evaluation (Liu et al., 2018, Wang et al., 2020a).

The majority of NLG systems have adopted crowd-sourcing platforms such as Amazon Mechanical Turk (AMT) for conducting human evaluation (Deriu et al., 2021). However this has not become the first choice in evaluating artificial danmu comments, the existing work (Ma et al., 2019, Wang et al., 2020b, Chaoqun et al., 2020, Zhang et al., 2020) has not provided detailed information regarding the source of the annotators. In our view, evaluating danmu comments requires annotators to be native speakers of Chinese and have extensive experiences in browsing danmu videos. These requirements are fundamental for establishing human evaluation of danmu commenting systems but are difficult to be satisfied in AMT. Naturally, the number of annotators is generally lower in danmu commenting evaluation, we found a median of 3 annotators in previous work while this number is usually 5 in other NLG applications (Callison-Burch, 2009, Kryscinski et al., 2019) where annotators are crowd-sourced.

Other than automated live video commenting, some of the criteria in evaluating automated dialogue systems could be of interest to us since the tasks are similar to each other. We thus also look at human evaluation schemes in automated dialogue systems. For evaluating dialogue systems, human annotators are typically asked to consider a much broader set of criteria such as:

- *Coherence*: Whether the response is coherent in the context of the surrounding dialogue. (Wu et al., 2019)
- *Appropriateness*: Whether the response is appropriate in grammar, topic, and logic. (Young et al., 2018)
- *Engagement*: Whether responses are engaging to user. (Zhang et al., 2018)
- *Informativeness*: Whether the response provides new information and knowledge in addition to the post. (Young et al., 2018, Tian et al., 2019, Zhu et al., 2019b, Wu et al., 2019, Zhang et al., 2020)

Our Approach We keep the two evaluation criteria (Fluency and Relevancy) used in (Ma et al., 2019) and (Wang et al., 2020b) for our human evaluation. A small adjustment is made regarding the Fluency of danmu comments. In the experimental scenarios of the danmu commenting task, the evaluated text are sourced from social media comments, which contain internet slang and tend to be informal and casual in terms of grammar, spelling and punctuation, in such cases some verbal imprecision is not necessarily to be criticised. Considering this, we specifically ask our annotators to slightly lower the standard of Fluency. We find the metric Correctness is not a major concern in our experimental scope where we place emphasis on increasing user engagement, therefore we do not include this metric in our human evaluation.

We also deliberate over the four metrics (listed above) that appear frequently in the evaluations of dialogue systems. In our case, even though sometimes danmu comments can form discussions, the dialogue structure of danmu comments are not explicit, thus a good danmu comment is not necessarily coherent or appropriate to every other contextual comment. For that reason, we do not consider Coherence and Appropriateness to be one of our evaluation focuses. Increasing user engagement is one of our initial research objectives, we include Engagement as a criterion in our previous human evaluation (see Section 5.4.2). However, in the current evaluation scope, we decided to include Informativeness instead of keeping using Engagement. One reason for this replacement comes from the annotator's feedback in the previous human experiment, all three annotators suggested that the metric Engagement is sometimes ambiguous, which makes them less confident in evaluation. As an alternative, we find the phrase Informativeness to be more objective and clear in evaluating danmu comments. It indicates the degree of comment novelty and can also reflect user engagements to some extent, since the attractiveness of a danmu comment mainly comes from the new information it contains.

Finally, considering that our danmu generation system is guided by the density prediction module, another evaluation focus will be around the effectiveness of the danmu insertion locations, thus we include "Timeliness: is the insertion point of the comment adequate?" as a new assessment dimension.

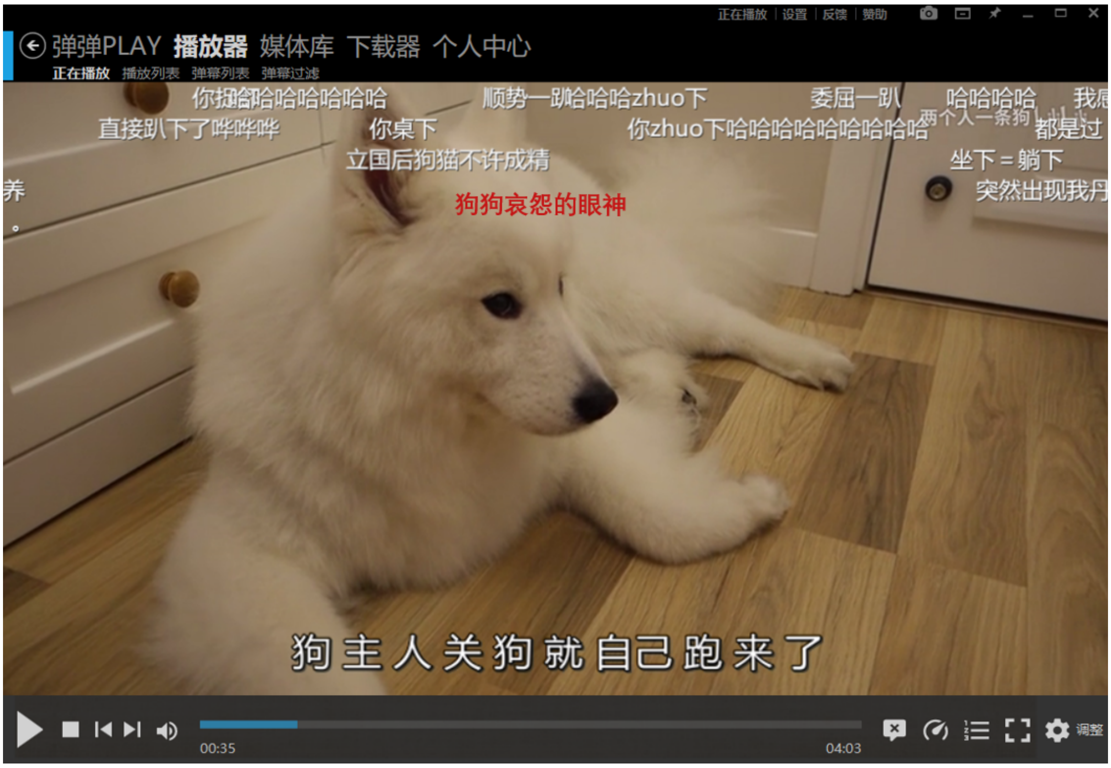


Fig. 8.1 The user interface of Dandan Play.

8.2 Human Evaluation

In this section, we present the detailed configurations our human evaluation experiment, which include the evaluation interface, annotators details and evaluation data.

8.2.1 Evaluation Setup

In our previous human evaluation, Section 5.4.2, the danmu comments to be evaluated and the corresponding videos are presented to the annotators separately. This isolates the annotators, from the real viewing experiences of danmu video websites. In previous work (Ma et al., 2019, Wang et al., 2020b), it is only mentioned that the comments are presented to the annotators and there is no further indication on whether or how the video content is presented to the annotators.

In this experiment, we attempt to simulate the real danmu website scenario, where annotators should see comments floating through the video playback screen. The first choice for such a platform would be the original Bilibili itself which introduced danmu, however, as the dataset was collected back in 2017, the original videos posted on Bilibili are already filled with newer danmu comments and some of the original videos are not accessible anymore. We thus decided to use a third-party danmu video player “Dandan Play” [1], which has a similar video interface to the Bilibili platform and

[1] https://www.dandanplay.com/

supports customized danmu lists uploads. The user interface of Dandan Play is shown in Figure 8.1. The comments that need to be evaluated are highlighted in red and the contextual comments in white for the convenience of the annotators.

We recruited three native Mandarin speakers with frequent danmu video viewing experiences as the annotators for the experiment, they are between 23 and 27 years old, consist of two males and one female. Typically, a systematic human assessment will require a larger group of annotators. We didn't recruit more annotators as the reported numbers of human evaluators in the literature of danmu comment generation (Ma et al., 2019, Zhang et al., 2020, Wang et al., 2019) are generally below 5. (Additionally, during the time of Covid it is more difficult to hire desired annotators.) In the experiment, they are required to watch the danmu video and rate our generated comments set from the above four criteria scaling from 0 to 5.

The data used in the experiment consists of 50 videos that are randomly sampled from the test set. For each video, among the system outputs, we selected 10 generated comments from 10 video-displaying timestamps associated with the highest predicted danmu density scores in the video timeline. Thus each annotator has a total of 500 danmu comments to evaluate per experiment.

8.2.2 Experimental Scenarios

Our evaluation is built around the following three cold start scenarios:
Unified (p=0%). In this scenario, the annotators are presented with comments generated from our unified Multi-Task framework in the complete cold start scenario ($p = 0\%$). No other danmu comments are presented to the annotators.
Unified (p=5%). The presented comments have been generated from our unified Multi-Task framework at $p = 5\%$. In addition, the annotators are exposed to the $p\%$ earliest context comments (shown in white instead of red in the interface).
Human (p=5%). The danmu comments are real comments taken from the videos. The locations are sampled from the most commented video locations. To keep it consistent with our generated comments, we only sample comments corresponding to $p = 5\%$ and also show the $p\%$ earliest context comments (shown in white).

Importantly, annotators are not informed of whether the comments for review are real or automatically generated.

8.3 Results

We average the scores from the three annotators and report the results in Table 8.1. We also report the confidence interval for each of the evaluation criteria with a confidence level at 95%.

The results show that our danmu annotation system can generate relevant and informative danmu comments in our near cold start scenario ($p = 5\%$), and even outperforms human comments in terms of Relevancy and Timeliness. Here notably, our predicted locations have a higher Timeliness score than

Model	p	Relevancy	Timeliness	Informativeness	Fluency
Unified	0%	3.744 ± 0.021	3.847 ± 0.066	3.952 ± 0.045	4.536 ± 0.025
Unified	5%	**3.985** ± 0.031	**4.421** ± 0.071	4.11 ± 0.076	4.533 ± 0.022
Human	5%	3.766 ± 0.034	4.118 ± 0.062	**4.266** ± 0.065	**4.752** ± 0.022

Table 8.1 Human evaluation results of our unified system in two evaluation scenarios. 500 comments are evaluated in each scenario. Each comment is graded between 0 and 5, by the 3 reviewers with respect to: language fluency, relevance, timeliness appropriateness and informativeness. The human comments are also evaluated for reference.

the groundtruth even with a slightly higher confidence interval score (recall that reference comments are sampled from the most commented locations in the original danmu video). For the other two evaluating criteria Informativeness and Fluency, our results are still very close to human comments (only 3.65% and 4.6% lower). The scores decrease when the system moves from near cold start to the complete cold start scenario ($p = 0\%$), in almost all of the evaluating dimensions. For Relevancy and Informativeness, we observe a minor performance drop by 6.04% and 3.84% respectively. However, the score drops dramatically (by 12.37%) when considering the Timeliness appropriateness of the artificial comments. One exception is found in Fluency, in this evaluating criteria our system achieves relatively high scores with low confidence intervals regardless of the presence of contextual comments. To evaluate the consistency among the scores from different annotators, we calculate Spearman's correlation coefficients (Myers and Sirois, 2004) between any two annotators. The average coefficient score is 0.68, which demonstrates the reliability of our human evaluation results.

8.4 Analysis of Good and Bad Results

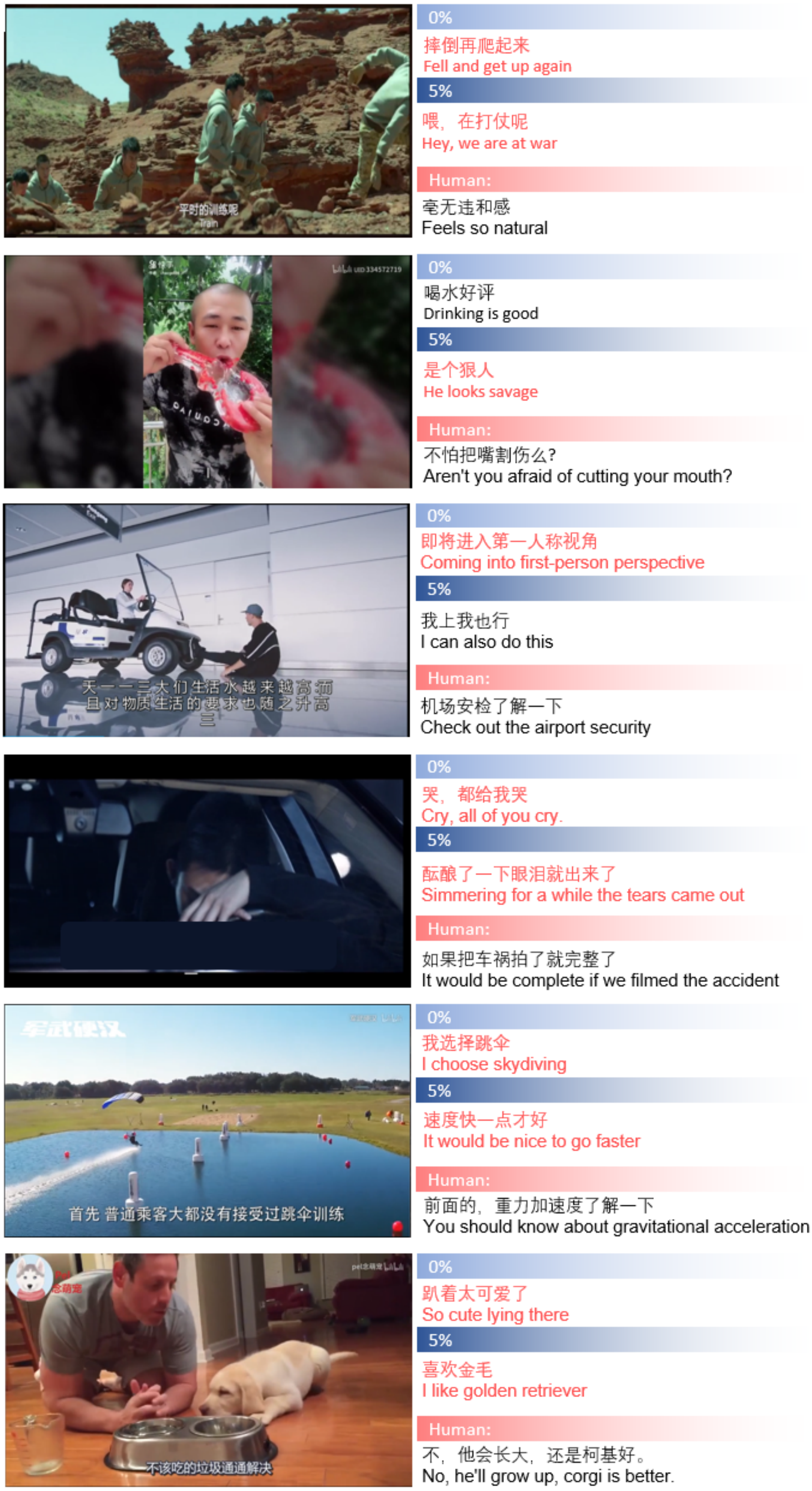
In order to study the performance of our system one step further, we look into concrete generation scenarios and present in Figure 8.2 several system outputs along with the corresponding video scenes. Specifically, we colour the generated comments in red if they have an average score (over four evaluation criteria) above 4.

A general observation over these user cases indicates that our system is able to generate fluent and informative comments even without analysing other historical user comments ($p = 0\%$). For example, in the left side lower case, where the video frame illustrates a person skydiving into a pool, our generated comments in each evaluation scenario are fairly accurate and relevant regarding the video scene. Sometimes our system is misled by the visual signal. In the top right case, where a man is crunching a glass bottle, the system mistakenly interprets it as a man drinking and produces a comment "Drinking is good" when taking zero contextual user comments as input. To perform an analysis of the situations where the system fails to generate relevant or meaningful comments, we select several cases that receive lower scores from annotators and present them in Figure 8.3. The comments labelled in green have an average score below 3. In the case on the top left, the system output ($p = 0\%$) is “A reminder in advance”, which is neither relevant to the context of the video (football match) nor informative. Some more instances of failure can be observed in the same figure, where the system output is not considered to be reasonable regarding video scenes and surrounding comments if given. Generic comments (“Hahahaha” in the bottom left with p=0% and “Ask for background music” in the bottom right with p=0%) that appear frequently in danmu videos will also be punished by the lack of informativeness and relevancy during human evaluation. There is also a case (bottom right of Figure 8.3) where the system output (“Oreo”) completely deviates from the video content. These poorly performing comments are generally unrelated to the corresponding video scenes, which potentially demonstrates the weakness of our system in correctly interpreting video signals.

One interesting finding is that the human comments are most likely responding to other previous danmu comments (*e.g.* human comment in the skydiving frame is talking to a previous comment in a conversation manner). While our reasonably performing artificial comments tend to be more relevant to the video content, since during the evaluation stage the system is only exposed to zero or little previous contextual danmu comments. This conversational / commentary difference makes the human comments vulnerable during the evaluation of Relevancy, since the original comments that they interact with may not exist in the context comment set which corresponds to only the earliest 5% of the original comments.

8.5 Conclusions

In this chapter, we proposed a scheme for comparative human evaluation of automated danmu creation against naturally occurring human danmu annotations of the same video content. Our proposed evaluation plan improves over the human evaluation schemes proposed in (Ma et al., 2019) and (Wang et al., 2020b), since our plan better reflects the goals of full simulation of human danmu creation. Instead of evaluating comments in a static interface, we stimulated the original user viewing interface using a third-party tool. The evaluation results of 50 videos from three annotators showed that the artificial comments outscored human comments when it comes to Relevancy and Timeliness. The scores over four evaluation criteria consistently indicate that our system is able to create high-quality comments of comparable quality for the human assessor to real ones.



0%
摔倒再爬起来
Fell and get up again
5%
喂，在打仗呢
Hey, we are at war
Human:
毫无违和感
Feels so natural

0%
喝水好评
Drinking is good
5%
是个狠人
He looks savage
Human:
不怕把嘴割伤么?
Aren't you afraid of cutting your mouth?

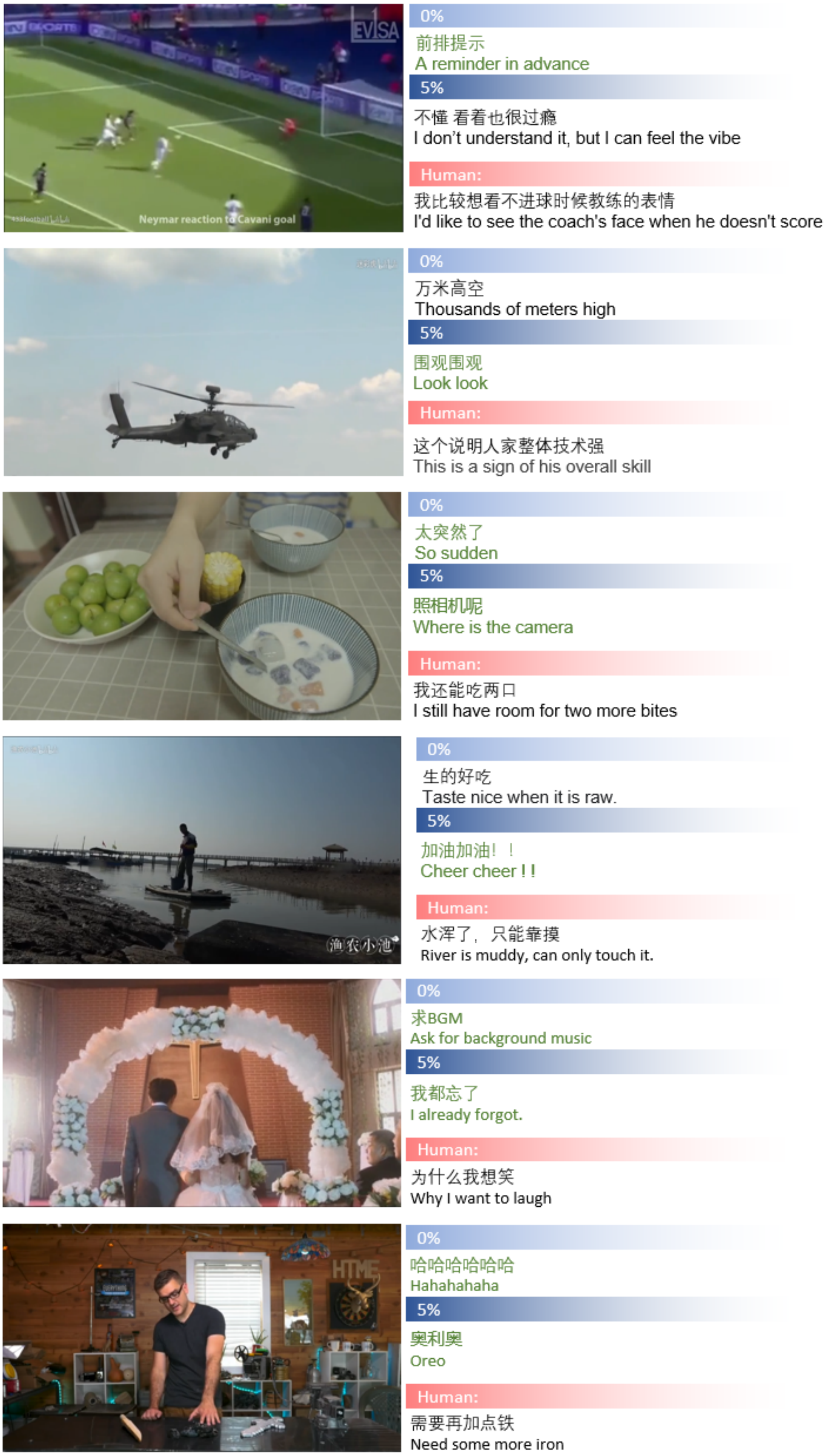
0%
即将进入第一人称视角
Coming into first-person perspective
5%
我上我也行
I can also do this
Human:
机场安检了解一下
Check out the airport security

0%
哭，都给我哭
Cry, all of you cry.
5%
酝酿了一下眼泪就出来了
Simmering for a while the tears came out
Human:
如果把车祸拍了就完整了
It would be complete if we filmed the accident

0%
我选择跳伞
I choose skydiving
5%
速度快一点才好
It would be nice to go faster
Human:
前面的，重力加速度了解一下
You should know about gravitational acceleration

0%
趴着太可爱了
So cute lying there
5%
喜欢金毛
I like golden retriever
Human:
不，他会长大，还是柯基好。
No, he'll grow up, corgi is better.

Fig. 8.2 System outputs in two evaluation scenarios (0% or 5% comment density) with corresponding video frame. Artificial comments above 4 (averaged over 4 evaluation dimension) are labelled in red.



0%
前排提示
A reminder in advance
5%
不懂 看着也很过瘾
I don't understand it, but I can feel the vibe
Human:
我比较想看不进球时候教练的表情
I'd like to see the coach's face when he doesn't score

0%
万米高空
Thousands of meters high
5%
围观围观
Look look
Human:
这个说明人家整体技术强
This is a sign of his overall skill

0%
太突然了
So sudden
5%
照相机呢
Where is the camera
Human:
我还能吃两口
I still have room for two more bites

0%
生的好吃
Taste nice when it is raw.
5%
加油加油！！
Cheer cheer ! !
Human:
水浑了，只能靠摸
River is muddy, can only touch it.

0%
求BGM
Ask for background music
5%
我都忘了
I already forgot.
Human:
为什么我想笑
Why I want to laugh

0%
哈哈哈哈哈哈
Hahahahaha
5%
奥利奥
Oreo
Human:
需要再加点铁
Need some more iron

Fig. 8.3 Cases where the system is unable to generate reasonable and high quality comments. Artificial comments scored below 3 (averaged over 4 evaluation dimension) are labelled in green.

Chapter 9

Conclusions and Future Research Directions

Danmu comments establish firm connections between video viewers, this emerging new feature engages hundreds of millions of users in rich community discussions and also improves users viewing experience. Videos with many danmu comments stand a higher chance of being recommended or searched, and thus naturally attract more viewers, this finding motivates us to improve the popularity and the viewing experience of a danmu video by contributing automated danmu comments and this forms the basis of our research objective. In this chapter we review the research questions individually and present our findings with respect to addressing each of these, following this we consider directions for potential future research extending and developing beyond the work presented in this thesis.

9.1 Review of the Research Questions

9.1.1 Establishing a Baseline for Danmu Comments Generation

To begin with, we conducted a comprehensive literature review around the topics of natural language generation(NLG), multimedia processing and social content analysis. During our investigation of the areas of NLG and multimedia processing, we drew a key conclusion that the Transformer architecture has gradually replaced LSTM structure and is frequently considered as a universal solution for tasks such as text processing, text generation, video processing and even multi-modal fusion. This finding provided valuable guidance to us in selecting neural architectures. As we inspected the relevant work in social content analysis, some work on social media highlight analysis caught our attention, where the general goal is to detect the most interesting and important clips of a video. Coincidentally, this task provided us with a new direction in automated danmu commenting, as in danmu videos, these highlights could potentially serve as preferred locations for automated commenting.

After reviewing the general fields relevant to our research goal, we looked at existing literature on automated danmu commenting. In a danmu video, the video visual signals and the viewer comments are aligned with each other along the video timeline, which means the video content and the danmu comments within a short video analysis window are closely correlated. Intuitively, multi-modal fusion techniques could be leveraged in combining and transforming the features of these two time-synchronized modalities. Ma et al. (2019) leveraged this idea and proposed to utilize both the existing danmu history and the visual information in automated danmu commenting. They fused the multi-modal signals with Transformer blocks. This work served as a suitable starting point for us, since they provided a clear task definition, a publicly available danmu video dataset, an automated evaluation scheme and baseline network architectures. It has also become a benchmark for many follow-up studies such as (Zhang et al., 2020, Chaoqun et al., 2020). When we attempted to reproduce the results from (Ma et al., 2019), we discovered several issues in their codebase and dataset which led to a performance mismatch between their published results and our attempt to replicate them. In order to provide a reproducible implementation for later research on the danmu commenting task, we re-implemented the transformer network of LiveBot in (Wu et al., 2020b) using the OpenNMT (Klein et al., 2017) open-source neural machine translation framework. We also expanded their released dataset by doubling its size.

9.1.2 RQ1: The Video Cold Start Problem for Comment Generation

We noted that most of the existing literature (Ma et al., 2019, Wang et al., 2020b, 2019, Lv et al., 2019a) on automated danmu commenting had focused on the analysis of videos that already have many comments. This is however probably not the most critical scenario for automated danmu generation as these videos are already popular. Similar to the "cold start problem" in recommender systems, the real issue faced by content creators is that videos need many danmu comments to start attracting traffic. To systematically address the cold start problem in danmu video commenting, we proposed our first research question:

[RQ1] Can we automatically create meaningful comments for less-commented or even uncommented videos?

We attempted to address this research question by investigating different levels of cold start scenarios in danmu videos. We approached the problem by generating comments from videos with various comment densities and proposed a multi-modal fusion network, that includes the processing of video frames, audio soundtrack, already published comments and video subtitles. To handle different comment density scenarios in the dataset, we adopted a multi-density learning strategy during training of the system and performed extensive experiments on the expanded danmu video dataset. Our results demonstrate the advantage of our method over the state-of-the-art in solving the cold video start problem. Apart from the automatic evaluation, we ran a human evaluation where the generated comments are compared with human comments in terms of their relevancy, engagement and fluency. Analysis of the results of our human evaluation suggested that our artificial comments are competitive in all evaluating dimensions against human comments.

According to the results from both automatic metrics and human evaluation, we reached the conclusion that our system can generate meaningful comments for videos with much fewer or zero comments. This finding indicates that at this stage we have successfully addressed the first research question.

9.1.3 RQ2: Where to Publish in the Video Timeline?

We noticed that the collective distribution of danmu comments over the video timeline exhibits interesting patterns and can represent the viewer's interest in different parts of the video. Densely commented scenes indicate a high level of user engagement thus can be seen as highlights of danmu videos. These danmu video highlights provide a solution to our second research question:

[RQ2] Can we automatically identify appropriate locations in video timelines to insert comments?

As a step towards the second research question, we first aimed at detecting potential highlights in the danmu videos before they start to attract traffic. Specifically, we used comment density as an indicator of video popularity and classify a densely commented video scene as a highlight. Based on such a setting, we formed a novel task of danmu density prediction. We re-used the transformer-based video encoder in this task and compared the performance against an LSTM baseline method. Our extensive experiments on the expanded danmu video dataset revealed three major conclusions: first, our proposed system constantly outperforms the baseline method, which demonstrates the effectiveness of the transformer architecture and our multi-modal fusion strategy. Second, we noted that in other video highlight detection tasks, visual inputs usually dominate the model performance (Godi et al., 2017, Xiong et al., 2019). However, the results from the ablation investigation indicated that the audio soundtrack is the most contributed modality. Third, we manually examined a few examples of the model output and found that our density predictions, even in a cold start scenario, were correlated with the groundtruth distribution.

Generally speaking, our proposed system produced accurate predictions on detecting danmu highlights with an mAP score of 57.69% in the complete cold start scenario. To this extent, we partly solved the fourth research question of finding appropriate locations to insert comments.

We then kept exploring the second research question by connecting the density prediction module with the danmu commenting module and considered the actual scenario of commenting at predicted highlights. We proposed to solve the problems of generating danmu comments and predicting where to insert them in the video with a single unified Multi-Task framework. In this framework, the two tasks were trained simultaneously with a shared encoder in an end-to-end manner. The two subsystems were evaluated separately against state-of-the-art baseline methods. The results showed that the Multi-Task approach consistently outperforms the single-task baselines, this findings supported our proposal that the information learned from modelling danmu comment distribution can benefit the comment generation module and vice versa. We also investigated the influences of commenting locations on the performance of the danmu generation module and notice that the unified framework performed better when commenting on its predicted highlights.

According to the above analysis, we successfully demonstrated that it is possible to predict danmu highlights and use them as preferred locations to insert comments, therefore solving the second research question.

9.1.4 RQ3: How Close Are We to Human Commenting?

In order to perform a comprehensive evaluation of the unified system, we raised our last research question:

[RQ3] How do our automatically created comments compare to human comments for the same videos?

We conducted a novel danmu commenting human evaluation with consideration of the objectives of predicting both content and the insert locations for the comments. This human evaluation includes four carefully selected criteria which differs slightly from the human evaluation proposed in (Ma et al., 2019, Wang et al., 2020b) by adding a Timeliness dimension, which measures the appropriateness of the commenting locations. We stimulated the original user viewing interface using a third-party tool in order to provide annotators with real danmu video viewing experiences. Artificial comments and human comments from 50 videos in the test set were manually evaluated by three experienced annotators.

For the evaluation of the danmu commenting system, we believe that the human evaluation plan we proposed delivers precise measurements over the quality of automated comments regarding multiple dimensions and makes a considerable improvement over existing human evaluation schemes for the same task. We conclude that we have found a suitable evaluation scheme for measuring the performance of generated comments. The evaluation results showed that when compared to human comments, the automated comments are more relevant to the video and timely appropriate. Besides this, human and automated comments have close performance in terms of informativeness or fluency. These findings indicate that our system can create high-quality comments of comparable quality to real ones.

9.2 Future Directions

In this section, several directions are presented for potential future research extending our current work on danmu video commenting.

Using Pre-trained Models For the moment, we are able to successfully generate meaningful and relevant comments at desired locations by analysing danmu videos of various comment densities. One possible future direction is to acquire external knowledge from pre-trained models, so far we only applied pre-trained visual models to enhance the representations of video frames. Models pre-trained on other modalities are very likely to improve the performance of a danmu commenting system. For example, pre-trained language models like BERT (Devlin et al., 2019) and its variations provide high-quality sentence-level text embeddings. With simple fine-tuning processes, these models can be applied in almost all NLP tasks and recently in NLG tasks (Zhu et al., 2020a, Wu et al., 2020a). In our case, we could replace the input text vectors with pre-trained text embeddings. Although the most examined scenarios are based on English corpus, in recent work, text models resources pre-trained on Chinese corpus (Cui et al., 2021) are becoming increasingly prevalent, and have the potential to be leveraged in improving the performance of automated danmu commenting systems. Aside from text models, Vision-Language models (*e.g.* CLIP (Radford et al., 2021) or DALL-E (Ramesh et al., 2021)) deliver a more efficient solution to multi-modal tasks (Huo et al., 2021). Ideally, a well pre-trained vision-language model could almost replace the whole video encoder in our framework, and is very possible to improve the performance with the knowledge acquired from a significantly larger multi-modal dataset.

Recent Advances in Multimedia Processing The multimedia analysis is a fast-evolving field and some of the outlined methods (e.g. multimedia feature extraction, text feature extraction, multi-modal fusion) were published at the beginning of the thesis work and are likely to have progressed significantly. By adapting to the recent advances of multimedia processing, our proposed danmu commenting system can be improved periodically. One particular field that may bring considerable improvement is audio processing, as audio soundtrack is proved to be a crucial modality in our system (see Section 6.3.5). The audio feature we used (MFCCs) is not optimized for the danmu commenting taks. Replacing it with more advanced features (*e.g.* mel-spectrogram) will very likely produce positive gains.

Leveraging Video Meta-Data Apart from gaining knowledge from external sources, another future direction in improving the model performance can be explored by leveraging video meta-data. Some video meta-data like video titles, tags and categories are worth analysing in the danmu commenting task as they are directly related to the video content and sometimes danmu comments (He et al., 2017). As we discussed in Chapter 6, as we analyse a video, the system will sometimes be misled by a limited video analysis window without seeing the global interpretation of that video. In this way, the video encoding module may fail to capture the correct features from input signals, therefore, misinterpreting the video segment. This weakness can be potentially improved by integrating more video meta-data into model inputs. These video meta-data could provide the video encoding module with fairly accurate contextual global information when analysing each individual video clip and therefore mitigate the risk of misinterpretation.

Highlights Detection Achievements from our studies in automated danmu commenting can also benefit other applications like danmu video highlights detection, which is crucial for enhancing the efficiency of video browsing on social media platforms. In our unified system, the danmu density prediction module can detect highlights in danmu video timelines and use them as locations of automated comments. Beyond this, the density prediction module can provide previews of danmu videos by navigating platform users to recommended highlighted scenes of individual videos.

Danmu Comments as Dialogue Another future work can be conducted around detecting the dialogue structure in danmu comments. Although we observed that danmu comments can sometimes form discussions, the dialogue structure of these discussions is not explicit. In the danmu commenting task, we form the training data by associating the target comment with several random surrounding contextual comments that are close in the video timeline, this process is based on a weak assumption that the publication of a comment is motivated by its neighbour comments (Ma et al., 2019). However, the real situation is often more sophisticated, for a given target comment, comments that it interact with may have been filtered out during random selection or this target comment is simply commenting on the video content without engaging any previous comments. Therefore, in our current setting, the target comments are not necessarily closely correlated with the input contextual comment and this setting subsequently create noise in the training set. This situation could be significantly improved if the dialogue structure underlying the commenting history were to be automatically detected and later used in formulating training sets of the commenting task.

Going Live Currently, our work remains at an experimental stage, however, such an automated commenting system has the potential to function in real-time. For danmu video websites like Bilibili, our system can help the content creator in increasing the user engagement of their freshly uploaded videos by contributing comments at appropriate places. From the viewer aspect, meaningful and informative comments delivered from our system can improve the viewing experience. Other than danmu video platforms, live streaming websites like Douyu or Twitch are also appropriate platforms to deploy the automated commenting system. By analysing the video streams and previous chats of users, the system would be able to literally communicate with viewers or streamers in real-time if given adequate computational resources. Deploying the system online also provide us with possible channels for evaluating the danmu commenting system. For example, A/B testing is a useful method for measuring user engagement and satisfaction. This could be implemented by having two collections of danmu videos and only publishing automated comments on the videos in one of them, then the user engagement can be measured by monitoring the growth of video statistics like the number of views or comments.

The Risks of Inappropriate Language We believe that the automated commenting systems have a great prospect in improving user experiences of video websites. However, we also face potential risks when bringing an automated commenting system live. A notable one is that the outputs generated from end-to-end neural network are highly unpredictable, which suggests that the system may publish hazard speech if it operates without supervision. The incident of "Tay" [1] (a social chatbot designed

[1] https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/

by Microsoft) is a good illustration. Tay was shut down after releasing racist and sexually-charged messages on Twitter.

To avoid incidents like Tay, one possible solution is to regulate the system by having a classifier (Niu et al., 2020) filtering out inappropriate speech before comments are published. From another perspective, densely aggregated artificial comments may, by contrast, lower the user experience. Therefore, the publishing rate of the system should be carefully controlled in a reasonable range.

Transparency and Potential Misuse Publishing machine comments on danmu videos may lead to a number of ethical issues as the comments will be exposed to platform users (both viewers and content creators) and produce further interactions. Before going live, the transparency of the artificial content should be determined carefully: whether the users should be aware of which comments are artificial; how this information should be flagged to users to make them aware of it, and how the user should be able to control them.

On the other aspect, ethical issues may arise from the potential misuse of the proposed commenting system. Experience shows that negative commenting and content can produce traffic, and countermeasures like the previously mentioned comment filtering methods need to be prepared to prevent companies from using the technologies to spam malicious or controversial comments through a different approach to training.

Data Bias Data bias is another problem that will eventually emerge from the process of the deployment of the commenting system. It refers to the fact that the training set may not reflect the realities of the environment where a model will run (Garrido-Muñoz et al., 2021). During the formation of the dataset, we attempt to eliminate bias in the dataset by generalizing the topics included in videos. However we notice that danmu video platforms are constantly changing and evolving, these changes do not only have an impact on video content but also on danmu comments. Some types of video or user commenting habits may have been very popular in 2017, but are no longer seen in the next year. Consequently, the model trained on our danmu video dataset may not perform well in danmu videos that are published recently and the comments it produces may also seem “outdated” to viewers nowadays. Intuitively, the addition of a more recently collected dataset could eliminate the bias caused by changes over time, to guarantee a stable performance of an online commenting system, it is necessary to retrain the model on a regular basis.

Explainable AI An online commenting AI system is a sensitive application for its direct interaction with humans and the potential risks associated with it as we mentioned above. For this reason, the need for explainable AI is particularly acute in this area (Reiter, 2019, Xu et al., 2018). Explainable AI refers to a set of processes and methods that allows human users to comprehend the mechanism and trust the results and output created by AI systems. Our system is built upon deep learning models in an end-to-end manner, the running status of a neural network module for either danmu commenting or density prediction is turned into what is commonly referred to as a “black box" that is impossible to interpret. Explainable AI could possibly help us by revealing the mechanism behind deep learning calculation processes, or potentially answering questions like “which previous comment is the system currently replying to?” or “why is this part of the video a highlight to viewers?”. These explanations would help us in understanding the behaviours of AI agents, this would enable the system to be gradually improved in a controlled environment.

Bibliography

T. Afouras, J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Deep audio-visual speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

N. Agarwal, R. Gupta, S. K. Singh, and V. Saxena. Metadata based multi-labelling of youtube videos. In *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, pages 586–590. IEEE, 2017.

J. Alammar. The illustrated transformer. http://jalammar.github.io/illustrated-transformer/, 2020.

S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

H. Aradhye, G. Toderici, and J. Yagnik. Video2text: Learning to annotate video content. In *2009 IEEE International Conference on Data Mining Workshops*, pages 144–151. IEEE, 2009.

J. Assfalg, M. Bertini, A. Del Bimbo, W. Nunziati, and P. Pala. Soccer highlights detection and recognition using hmms. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 825–828. IEEE, 2002.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *In Proceedings of the 3rd International Conference on Learning Representations*, 2015.

S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on Intrinsic and Rxtrinsic Rvaluation Measures for Machine Translation and/or Summarization*, pages 65–72, 2005.

O. Barnich and M. Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, 2010.

R. Barzilay and L. Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational 2004 (HLT-NAACL)*, pages 113–120. ACL, 2004.

H. Bhuiyan, J. Ara, R. Bardhan, and M. R. Islam. Retrieving youtube video by sentiment analysis on user comment. In *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 474–478. IEEE, 2017.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of Machine Learning Research*, 3:993–1022, 2003.

P. F. Brown, V. J. Della Pietra, P. V. Desouza, J. C. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–480, 1992.

O. Caglayan, W. Aransa, Y. Wang, M. Masana, M. García-Martínez, F. Bougares, L. Barrault, and J. van de Weijer. Does multimodality help human and machine for translation and image captioning? In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 627–633. ACL, 2016.

D. Cai and H. Zhao. Neural word segmentation learning for Chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics)*, pages 409–420. ACL, 2016.

D. Cai, H. Zhao, Z. Zhang, Y. Xin, Y. Wu, and F. Huang. Fast and accurate neural word segmentation for Chinese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 608–615. ACL, 2017.

C. Callison-Burch. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 286–295, 2009.

J. Carreira, E. Noland, C. Hillier, and A. Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019.

D. Chaoqun, C. Lei, M. Shuming, W. Furu, Z. Conghui, and Z. Tiejun. Multimodal matching transformer for live commenting. In *In Proceedings of the 24th European Conference on Artificial Intelligence*, 2020.

X. Chen, X. Qiu, C. Zhu, and X.-J. Huang. Gated recursive neural network for chinese word segmentation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1744–1753, 2015a.

X. Chen, X. Qiu, C. Zhu, P. Liu, and X.-J. Huang. Long short-term memory neural networks for chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206, 2015b.

K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111. ACL, 2014.

K. M. Colby. *Artificial paranoia: A computer simulation of paranoid processes*, volume 49. Elsevier, 2013.

R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, 2003.

Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514, 2021.

A. A. L. Cunha, M. C. Costa, and M. A. C. Pacheco. Sentiment analysis of youtube video comments using deep neural networks. In *International Conference on Artificial Intelligence and Soft Computing*, pages 561–570. Springer, 2019.

X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Qiu Chen. Temporal context network for activity localization in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5793–5802, 2017.

Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–335, 2017.

J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender Systems*, pages 293–296, 2010.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

J. Deriu, A. Rodrigo, A. Otegi, G. Echegoyen, S. Rosset, E. Agirre, and M. Cieliebak. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54(1):755–810, 2021.

J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1370–1380. ACL, 2014.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186. ACL, 2019.

M. A. Di Gangi, M. Negri, R. Cattoni, D. Roberto, and M. Turchi. Enhancing transformer for end-to-end speech-to-text translation. In *Machine Translation Summit XVII*, pages 21–31. European Association for Machine Translation, 2019.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *In Proceedings of the 8th International Conference on Learning Representations*, 2020.

J. Du and A. Way. Pinyin as subword unit for chinese-sourced neural machine translation. In *AICS*, pages 89–101, 2017.

M. Duncan, A. Pelled, D. Wise, S. Ghosh, Y. Shan, M. Zheng, and D. McLeod. Staying silent and speaking out in online comment sections: The influence of spiral of silence and corrective action in reaction to news. *Computers in Human Behavior*, 102:192–205, 2020.

G. Galatas, G. Potamianos, and F. Makedon. Audio-visual speech recognition incorporating facial depth information captured by the kinect. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 2714–2717. IEEE, 2012.

I. Garrido-Muñoz, A. Montejo-Ráez, F. Martínez-Santiago, and L. A. Ureña-López. A survey on bias in deep nlp. *Applied Sciences*, 11(7):3184, 2021.

X. Ge, W. Pratt, and P. Smyth. Discovering chinese words from unsegmented text. In *Proceedings of the 22nd Annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 271–272, 1999.

D. Gkatzia. Content selection in data-to-text systems: A survey. *arXiv preprint arXiv:1610.08375*, 2016.

M. Godi, P. Rota, and F. Setti. Indirect match highlights detection with deep convolutional neural networks. In *International Conference on Image Analysis and Processing*, pages 87–96. Springer, 2017.

A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. Ieee, 2013.

L. Grossman. (list: 50 best websites 2013). 2013.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

M. He, Y. Ge, E. Chen, Q. Liu, and X. Wang. Exploring the emerging type of comment for online videos: Danmu. *ACM Transactions on the Web (TWEB)*, 12(1):1–33, 2017.

S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (icassp)*, pages 131–135. IEEE, 2017.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

M. Hu, J. Chen, and C. Shi. Three-dimensional mapping based on sift and ransac for mobile robot. In *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 139–144. IEEE, 2015.

P.-Y. Huang, F. Liu, S.-R. Shiang, J. Oh, and C. Dyer. Attention-based multimodal neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 639–645, 2016.

Y. Huo, M. Zhang, G. Liu, H. Lu, Y. Gao, G. Yang, J. Wen, H. Zhang, B. Xu, W. Zheng, et al. Wenlan: Bridging vision and language by large-scale multi-modal pre-training. *arXiv preprint arXiv:2103.06561*, 2021.

V. Iashin and E. Rahtu. Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 958–959, 2020.

J. Ive, P. Madhyastha, and L. Specia. Distilling translations with visual awareness. In *Proceedings of the 57th Annual Meeting of ACL*, pages 6525–6538. ACL, 2019.

G. Iyengar, H. J. Nock, and C. Neti. Audio-visual synchrony for detection of monologues in video archives. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 5, pages V–772. IEEE, 2003.

Y. Jiao, Z. Li, S. Huang, X. Yang, B. Liu, and T. Zhang. Three-dimensional attention-based deep ranking model for video highlight detection. *IEEE Transactions on Multimedia*, 2018.

W. Jin, Z. Zhao, M. Gu, J. Xiao, F. Wei, and Y. Zhuang. Video dialog via progressive inference and cross-transformer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP*, pages 2109–2118, 2019.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of EACL*, pages 427–431. ACL, 2017.

T. Jumneanbun, S. Sae-Lao, P. Paliyawan, R. Thawonmas, K. Sookhanaphibarn, and W. Choensawat. Rap-style comment generation to entertain game live streaming. In *2020 IEEE Conference on Games (CoG)*, pages 706–707. IEEE, 2020.

A. Kay. Tesseract: an open-source optical character recognition engine. *Linux Journal*, 2007(159):2, 2007.

Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR 2015*, 2014.

R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3294–3302, 2015.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the 2017 Conference on the Association for Computational Linguistics, System Demonstrations*, pages 67–72. ACL, 2017.

A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *International Journal of Computer Vision*, 50(2): 171–184, 2002.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

W. Kryscinski, N. S. Keskar, B. McCann, C. Xiong, and R. Socher. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551. ACL, 2019.

X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, pages 208–211. ACM, 2008.

Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

C. Li, J. Wang, H. Wang, M. Zhao, W. Li, and X. Deng. Visual-texual emotion analysis with deep coupled video and danmu neural networks. *IEEE Transactions on Multimedia*, 22(6):1634–1646, 2019.

J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202. ACL, 2016.

P. Li, W. Lam, L. Bing, and Z. Wang. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100. ACL, 2017.

J. Libovický, J. Helcl, M. Tlustý, O. Bojar, and P. Pecina. CUNI system for WMT16 automatic post-editing and multimodal translation tasks. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 646–654. ACL, 2016.

C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

C.-W. Liu, R. Lowe, I. Serban, M. Noseworthy, L. Charlin, and J. Pineau. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132. ACL, 2016.

S. Liu, H. Chen, Z. Ren, Y. Feng, Q. Liu, and D. Yin. Knowledge diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1489–1498, 2018.

B. Logan. Mel frequency cepstral coefficients for music modeling. In *In International Symposium on Music Information Retrieval*. Citeseer, 2000.

D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. Ieee, 1999.

J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

G. Lv, T. Xu, Q. Liu, E. Chen, W. He, M. An, and Z. Chen. Gossiping the videos: An embedding-based generative adversarial framework for time-sync comments generation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 412–424. Springer, 2019a.

G. Lv, K. Zhang, L. Wu, E. Chen, T. Xu, Q. Liu, and W. He. Understanding the users and videos by mining a novel danmu dataset. *IEEE Transactions on Big Data*, 2019b.

S. Ma, L. Cui, D. Dai, F. Wei, and X. Sun. Livebot: Generating live video comments based on visual and textual contexts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6810–6817, 2019.

A. B. Mabrouk and E. Zagrouba. Abnormal behavior recognition for intelligent video surveillance systems: A review. *Expert Systems with Applications*, 91:480–491, 2018.

T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur. Recurrent neural network based language model. In *11th Annual Conference of the International Speech Communication Association*, 2010.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 2013b.

G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

S. Min, F. Ali, and S. Steve. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 2014.

Y. Mroueh, E. Marcheret, and V. Goel. Deep multimodal learning for audio-visual speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2130–2134. IEEE, 2015.

L. Myers and M. J. Sirois. Spearman correlation coefficients, differences between. *Encyclopedia of statistical sciences*, 12, 2004.

D. Neimark, O. Bar, M. Zohar, and D. Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021.

C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, J. Sison, and A. Mashari. Audio visual speech recognition. Technical report, IDIAP, 2000.

H. Niu, J. Li, and Y. Zhao. Smartbullets: a cloud-assisted bullet screen filter based on deep learning. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–2. IEEE, 2020.

H. F. Nweke, Y. W. Teh, G. Mujtaba, and M. A. Al-Garadi. Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Information Fusion*, 46:147–170, 2019.

R. Ohbuchi and T. Furuya. Accelerating bag-of-features sift algorithm for 3d model retrieval. In *Proc. SAMT 2008 Workshop on Semantic 3D Media (S-3D)*, pages 23–30. Citeseer, 2008.

P. Owen. Our top 10 funniest youtube comments – what are yours? 2009.

A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1777–1784, 2013.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

F. Peng and D. Schuurmans. Self-supervised chinese word segmentation. In *International Symposium on Intelligent Data Analysis*, pages 238–247. Springer, 2001.

F. Peng, D. Schuurmans, S. Wang, and V. Keselj. Language independent authorship attribution using character level language models. In *Proceedings of the 10th conference on European Chapter of the Association for Computational Linguistics*, pages 267–274. Association for Computational Linguistics, 2003.

J. W. Pennebaker, M. E. Francis, and R. J. Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

V. Pérez-Rosas, R. Mihalcea, and L.-P. Morency. Utterance-level multimodal sentiment analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 973–982, 2013.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2227–2237. Association for Computational Linguistics, 2018.

P. Piccinini, A. Prati, and R. Cucchiara. Real-time object detection and localization with sift-based clustering. *Image and Vision Computing*, 30(8):573–587, 2012.

P. Qing and C. Chaomei. Video highlights detection and summarization with lag-calibration based on concept-emotion mapping of crowd-sourced time-sync comments. *arXiv:1708.02210*, 2017.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

T. Rahman, B. Xu, and L. Sigal. Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8908–8917, 2019.

A. M. Ramadhani and H. S. Goo. Twitter sentiment analysis using deep learning methods. In *2017 7th International annual engineering seminar (InAES)*, pages 1–4. IEEE, 2017.

A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

E. Reiter. Natural language generation challenges for explainable AI. In *Proceedings of the 1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence*, pages 3–7. ACL, 2019.

S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

A. Ritter, C. Cherry, and B. Dolan. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011.

A. B. Sai, A. K. Mohankumar, and M. M. Khapra. A survey of evaluation metrics used for nlg systems. *ACM Computing Surveys (CSUR)*, 55(2):1–39, 2022.

P. Schultes, V. Dorner, and F. Lehner. Leave a comment! an in-depth analysis of user comments on youtube. *Wirtschaftsinformatik*, 42:659–673, 2013.

F. Sebastiani and A. Esuli. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 417–422, 2006.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725. ACL, 2016.

G. Shao. Understanding the appeal of user-generated media: a uses and gratification perspective. *Internet research*, 2009.

R. P. Sharma and G. K. Verma. Human computer interaction using hand gesture. *Procedia Computer Science*, 54:721–727, 2015.

Z. Shen, J. Li, Z. Su, M. Li, Y. Chen, Y.-G. Jiang, and X. Xue. Weakly supervised dense video captioning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1916–1924, 2017.

Z. Shou, D. Wang, and S.-F. Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058, 2016.

Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5734–5743, 2017.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014a.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014b.

L. Specia, S. Frank, K. Sima'An, and D. Elliott. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 543–553, 2016.

S. Srivastava, M. Patidar, S. Chowdhury, P. Agarwal, I. Bhattacharya, and G. Shroff. Complex question answering on knowledge graphs using machine translation and multi-task learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3428–3439, 2021.

P. J. Stone, D. C. Dunphy, and M. S. Smith. The general inquirer: A computer approach to content analysis. 1966.

R. Su, X. Liu, L. Wang, and J. Yang. Cross-domain deep visual feature generation for mandarin audio–visual speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:185–197, 2019.

M. Sun and B. K. Tsou. Ambiguity resolution in chinese word segmentation. In *Proceedings of the 10th Pacific Asia Conference on Language, Information and Computation*, pages 121–126, 1995.

Z. Sun, M. Sun, N. Cao, and X. Ma. Videoforest: interactive visual summarization of video streams based on danmu data. In *SIGGRAPH ASIA 2016 Symposium on Visualization*, pages 1–8, 2016.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27, 2014.

P. Sykora, P. Kamencay, and R. Hudec. Comparison of sift and surf methods for use on hand gesture recognition based on depth map. *Aasri Procedia*, 9:19–24, 2014.

M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307, 2011.

H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

H. Tang, V. Kwatra, M. E. Sargin, and U. Gargi. Detecting highlights in sports videos: Cricket as a test case. In *2011 IEEE International Conference on Multimedia and Expo*, pages 1–6. IEEE, 2011.

Z. Tian, W. Bi, X. Li, and N. L. Zhang. Learning to abstract for memory-augmented conversational response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3816–3825, 2019.

Y. Ting, M. Tao, and R. Yong. Highlight detection with pairwise deep ranking for first-person video summarization. In *CVPR*, 2016.

D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the Conference. ACL. Meeting*, volume 2019, page 6558. NIH Public Access, 2019.

V. Tyagi and C. Wellekens. On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 1, pages I–529. IEEE, 2005.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. In *Proc. 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, page 125, 2016.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. pages 4534–4542, 2015.

O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

S. Walter, S. Scherer, M. Schels, M. Glodek, D. Hrabal, M. Schmidt, R. Böck, K. Limbrecht, H. C. Traue, and F. Schwenker. Multimodal emotion classification in naturalistic user behavior. In *International Conference on Human-Computer Interaction*, pages 603–611. Springer, 2011.

J. Wang, J. Liu, W. Bi, X. Liu, K. He, R. Xu, and M. Yang. Improving knowledge-aware dialogue generation via knowledge base question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9169–9176, 2020a.

T. Wang, X. Xu, Y. Yang, A. Hanjalic, H. T. Shen, and J. Song. Matching images and text with multi-modal tensor fusion and re-ranking. In *Proceedings of the 27th ACM international conference on Multimedia*, pages 12–20, 2019.

W. Wang, X. Yang, B. C. Ooi, D. Zhang, and Y. Zhuang. Effective deep learning-based multi-modal retrieval. *The VLDB Journal*, 25(1):79–101, 2016.

W. Wang, J. Chen, and Q. Jin. Videoic: A video interactive comments dataset and multimodal multitask learning for comments generation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2599–2607, 2020b.

Y. Wang, L. Zhou, J. Zhang, and C. Zong. Word, subword or character? an empirical study of granularity in chinese-english nmt. In *China Workshop on Machine Translation*, pages 30–42. Springer, 2017.

Y. Wang, S. Joty, M. Lyu, I. King, C. Xiong, and S. C. Hoi. VD-BERT: A Unified Vision and Dialog Transformer with BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3325–3338. ACL, 2020c.

J. Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

C.-S. Wu, S. C. Hoi, R. Socher, and C. Xiong. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 917–929. ACL, 2020a.

H. Wu, G. J. F. Jones, and F. Pitié. Response to livebot: Generating live video comments based on visual and textual contexts. *arXiv*, 2020b.

H. Wu, G. J. F. Jones, and F. Pitie. Knowing where and what to write in automated live video comments: A unified multi-task approach. In *Proceedings of the 2021 International Conference on Multimodal Interaction*, pages 619–627, 2021a.

H. Wu, F. Pitié, and G. Jones. Cold start problem for automated live video comments. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 54–62, 2021b.

H. Wu, F. Pitié, and G. J. Jones. Investigating automated mechanisms for multi-modal prediction of user online-video commenting behaviour. In *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2021c.

H. Wu, F. Pitié, and G. J. Jones. Investigating automated mechanisms for multi-modal prediction of user online-video commenting behaviour. In *2021 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6. IEEE, 2021d.

W. Wu, Z. Guo, X. Zhou, H. Wu, X. Zhang, R. Lian, and H. Wang. Proactive human-machine conversation with explicit conversation goal. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3794–3804, Florence, Italy, 2019. ACL.

Z. Wu and E. Ito. Correlation analysis between user's emotional comments and popularity measures. In *2014 IIAI 3rd International Conference on Advanced Applied Informatics*, pages 280–283. IEEE, 2014.

C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, and W.-Y. Ma. Topic aware neural response generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

B. Xiong, Y. Kalantidis, D. Ghadiyaram, and K. Grauman. Less is more: Learning highlight detection from video duration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1258–1267, 2019.

D. Xiong, Q. Liu, and S. Lin. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 521–528. ACL, 2006.

Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang. Highlights extraction from sports video based on an audio-visual marker detection framework. In *2005 IEEE International Conference on Multimedia and Expo*, pages 4–pp. IEEE, 2005.

C. Xu, Z. Yongfeng, A. Qingyao, X. Hongteng, Y. Junchi, and Q. Zheng. Personalized key frame recommendation. In *SIGIR*, 2017a.

C. Xu, W. Wu, and Y. Wu. Towards explainable and controllable open domain dialogue generation with dialogue acts. *arXiv preprint arXiv:1807.07255*, 2018.

J. Xu, T. Yao, Y. Zhang, and T. Mei. Learning multimodal attention lstm networks for video captioning. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 537–545, 2017b.

N. Xue. Chinese word segmentation as character tagging. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 8, Number 1, February 2003: Special Issue on Word Formation and Chinese Language Processing*, pages 29–48, 2003.

H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4633–4641, 2015.

N. Yang, S. Liu, M. Li, M. Zhou, and N. Yu. Word alignment modeling with context dependent deep neural network. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 166–175. ACL, 2013.

T. Yang. A research of the constructing of cultural field of danmu video website bilibili in china. *Science Innovation*, 6(3):164, 2018.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32, 2019.

S. Yao and X. Wan. Multimodal transformer for multimodal machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4346–4350, 2020.

T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, and M. Huang. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4584–4593, 2016.

J. Yu, Z. Zhu, Y. Wang, W. Zhang, Y. Hu, and J. Tan. Cross-modal knowledge reasoning for knowledge-based visual question answering. *Pattern Recognition*, 108:107563, 2020.

Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6281–6290, 2019.

A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency. Tensor fusion network for multimodal sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1114. ACL, 2017.

D. Zhang, H. Xu, Z. Su, and Y. Xu. Chinese comments sentiment classification based on word2vec and svmperf. *Expert Systems with Applications*, 42(4):1857–1863, 2015.

S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213. ACL, 2018.

Z. Zhang, Z. Yin, S. Ren, X. Li, and S. Li. Dca: Diversified co-attention towards informative live video commenting. In *CCF NLPCC*, 2020.

B. Zhao, S. Lin, X. Qi, Z. Zhang, X. Luo, and R. Wang. Automatic generation of visual-textual web video thumbnail. In *SIGGRAPH Asia 2017 Posters*, pages 1–2, 2017.

W. Zheng, Z. Jie, M. Jing, L. Jingjing, A. Jiangbo, and Y. Yang. Discovering attractive segments in the user-generated video streams. *Information Processing & Management*, 2020.

S. Zhou, X. Zeng, Y. Zhou, A. Anastasopoulos, and G. Neubig. Improving robustness of neural machine translation with multi-task learning. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 565–571, 2019.

C. Zhu, M. Zeng, and X. Huang. Multi-task learning for natural language generation in task-oriented dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1261–1266, 2019a.

J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, and T. Liu. Incorporating bert into neural machine translation. In *In Proceedings of the 8th International Conference on Learning Representations*, 2020a.

Q. Zhu, L. Cui, W.-N. Zhang, F. Wei, and T. Liu. Retrieval-enhanced adversarial training for neural response generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3763–3773, Florence, Italy, 2019b. ACL.

Z. Zhu, J. Yu, Y. Wang, Y. Sun, Y. Hu, and Q. Wu. Mucko: Multi-layer cross-modal knowledge reasoning for fact-based visual question answering. In *IJCAI*, pages 1097–1103, 2020b.