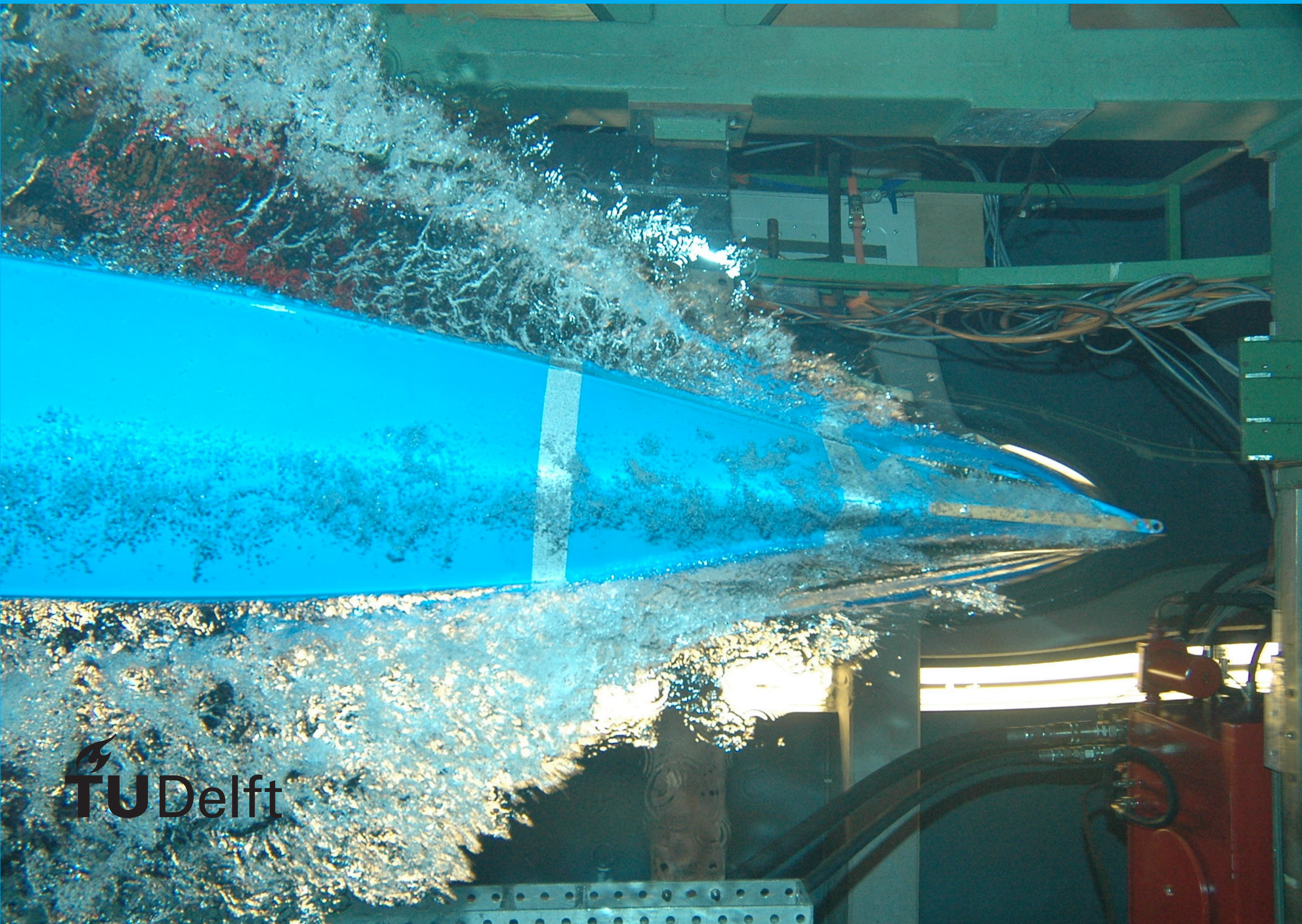
Video Captioning for the Visually Impaired

Fenglu Xu



Video Captioning

for the Visually Impaired

by

Fenglu Xu

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday September 11, 2024 at 11:00 AM.

Student number:	4579976	
Project duration:	December 6, 2018 – September 11, 2024	
Thesis committee:	Dr. Julián Urbano,	TU Delft, supervisor
	Dr. Odette Scharenborg,	TU Delft
	Dr. Jan van Gemert,	TU Delft
	Dr. Zhe Li,	TU Delft
	Benjamin Timmermans	IBM

This thesis is confidential and cannot be made public until September 11, 2024.

TU Delft

Preface

Before you lies the thesis work ”Video captioning for the visually impaired”. The goal of this thesis is to give an attempt to develop a video caption model, helping visually impaired people recognise their surroundings better. There are many people with visual impairment that meet difficulties in daily life in this world. The rapid developing deep learning technology is promising to help them. That is the motivation of this thesis. This thesis project was in collaboration with the IBM CAS Benelux and performed in Multimedia Computing Group at TU Delft for Koninkijke Visio, an institution that supports people with visual challenge and researches on what can help them.

This thesis is the final achievement of my study at Delft University of Technology. I believe I benefit from this journey even from a life-long perspective. In the aspect of academic, it provide an opportunity for me to dive deeper into the multidisciplinary area concerning nature language processing and computer vision. In the aspect of personal growth, this journey gives me a chance to develop a better understanding of myself.

I would like to thank Prof. Julian Urbano Merino and my company mentor Benjamin Timmermans firstly. Ben offered me this thesis project and Julian accepted to be my supervisor. During the whole process, they gave me not only the excellent supervision and guidance but also great patience and kindness. Furthermore, I would like to thank Dr. Zhe Li for his generous help and professional feedback. He was always willing to give useful advices when necessary. Finally, I would like to thank my family and friends for being there for me throughout this journey, in particular my mother. I couldn’t find the words to thank her enough. Her deep and quiet love is through everywhere in my life. Without her I would not be who I am today.

Fenglu Xu
Delft, September 2024

Contents

List of Figures

List of Tables

1 Introduction

1.1. Motivation

There are many people with visual impairment in the world. The statistics from the World Health Organisation (WHO) show that globally, more than 2.2 billion people have a vision impairment as of 2022. In the Netherlands, it was estimated that 311,000 individuals were visually impaired in 2008, comprising 77,000 who were blind and 234,000 with low vision. If current intervention measures remain unchanged, this number is projected to rise by 18%, reaching approximately 367,000 by 2020.[1]

Their life should be seen. Not as the majority of the non-blind community reckon, the blind people or the visually impaired can lead a normal life in their own way of doing things. Indeed, they meet difficulties in daily life because of different barriers like inaccessible public infrastructure, unreachable information etc.. Investigating those barriers can help us understand them, and using new technology might be helpful to eliminate barriers that hinder them from gaining independence.

Research on new technology for the visually impaired and blind people have become an important concern in the research area of assistive technologies. It traditionally focused on mobility, navigation, Internet of thing(IoT) and object recognition; but more recently on visual question answering (VQA), image/video caption and social interaction as well.

The rapid development of deep learning technology brings breakthroughs to many technologies that could be used to enhance the lives of people with visual impairment in the near future. Thanks to the availability of GPUs and CPU clusters as well as large amounts of training data, recent deep learning facilitated great success in computer vision (CV) and also gradually led to good performance on natural language processing (NLP) tasks like machine translation, etc. Video captioning, such a task that bridges CV and NLP together, has been revolutionised by deep learning as well, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs excel at analyzing visual imagery, such as object recognition, while RNNs have great influence over sequence modeling in machine translation and image captioning.

Video captioning is a promising research direction that may help blind and visually impaired people live much more independently. Video captioning is the process that "translates" a sequence of video frames into textual descriptions. It can be divided into two parts: understand the video contents visually and describe the video contents grammatically. In other words, it bridges two branches of AI, CV and NLP. Prevalent research work emerge, applied with the technology from Sequence-to-sequence model[157] to Transformer[154], in this field. It is a potential research area that may help visually impaired and blind people get verbal descriptions of their surroundings. While not limited to this, it can have more applications like video surveillance, human-robot interaction, and even help communication with speech-impaired people by "translating" sign language.

In the area of video captioning, the research community has integrated deep neural networks in various forms and has encouraged performance. Compared to template-based methods, deep learning methods are data-hungry, which allows video captioning to extend from fine-grained to open domain. In addition, deep learning relieves the burden of manually crafted features and manually designed language templates by multiple-layer neural network learning. Many deep learning methods are involved in developing video

[1] https://pubmed.ncbi.nlm.nih.gov/19995201/

captioning models like attention mechanism[24], deep reinforcement learning[140], generative adversarial networks[40], transformer[154], and so on.

All of these reasons aforementioned contribute to the motivation of this thesis. By developing a video caption model, we aim to help visually impaired people recognize their surroundings better. The research reported in this thesis extends the work of [157]. In this thesis, this model is enhanced to meet the needs of people with vision challenges. Extensive experiments demonstrate the effectiveness and interpretability of the model designed. The results are reported in the following chapters.

1.2. Research questions

The main research question of this research thesis is defined as follows:

How to develop a video captioning model for the visually impaired?

This main question can be split into six sub-questions, and these questions are answered in the following chapters.

1. What techniques can be used to build a video caption model for the visually impaired is investigated.
2. What are the requirements for video captioning special from the visual impaired?
3. How can we improve the performance of the baseline video caption model? (The baseline model is the S2VT model [157] with an attention layer)

   Because of the results found from the user study, more detailed research questions come up. Logically, these questions should be derived from the survey results presented in Chapter 3. However, to enhance the clarity of the thesis's structure, we will address them here in advance.

4. How can we design the video caption model to be more sensitive to actions?
5. How can the video caption model generate a sentence with good readability?
6. How to reduce the latency of the video caption model?

1.3. Thesis contribution

In summary, our contributions are four-fold.

1. Literature review. We present a comprehensive literature review on the video captioning task, with a particular focus on deep learning methods. Additionally, we summarize assistive solutions aimed at individuals with visual challenges, particularly those based on smartphone technology.
2. Survey. We conducted a survey of young visually impaired individuals at the Visio organization. For comparison, we also included senior participants aged over 60 in this survey. Based on the findings from the survey, we summarized several key insights that can guide and assist others researching the same topic.
3. Model. We propose a video captioning model tailored for individuals with visual impairments, grounded in an encoder-decoder framework that includes modifications to the temporal attention mechanism and specialized features. This model ultimately meets the requirements identified in our user study for the target user group.
4. Evaluate the model by readability. We employ readability metrics to evaluate the captions generated by the model. To our knowledge, no other work has utilized readability metrics for model assessment. This approach offers a novel perspective on evaluating model performance.

1.4. Thesis outline

The thesis is organized as follows.

1. We first proceed to a literature review of the different fields concerned with our research questions (the contemporary research work of video captioning and assistive technology for the visually impaired mainly) to answer the first sub-questions (Chapter 2).

2. Then we tackle the second subquestion (RQ2): As we aim at designing a video caption model for the visual impaired, we conducted several questionnaires with the visually impaired. Based on the questionnaires, we formed a list of requirements, and interesting insights are also included in Chapter 3.
3. Afterwards, In Chapter 4, we focus on the model design with respect to research questions 3-5 (which are directly tailored to the visually impaired people's needs), from sampling strategy and feature extraction to encoder-decoder architecture.
4. In the next chapter, we analyse dataset used firstly and then we give details of how we prepare the data and the procedure we train the models (Chapter 5).
5. Chapter 6 is experiment result and analysis where the details of four ablation study are discussed for remaining four research subquestions respectively. We compare the performance of our model with different add-on components like adjusted attention and various pertained feature models. Then, we present the model performance on different categories of video in order to investigate the effectiveness of the solution we give to Research Question 3: How can we design the video caption model more sensitive to actions? In the following subsection, we introduce several metrics to measure the readability of our models. In the last subsection, the latency of the model with different add-on components is contrasted. Furthermore, Research Question 5: How to reduce the latency of the video caption model? is addressed.
6. In the next two Chapters, we discuss the limitations and future work first (chapter 6) and then conclude the overall results with respect to research questions.
7. This thesis was started in 2018 and has been interrupted several times over the years due to personal health reasons. This year, I was able to continue working on this thesis again. Since a long time has passed, I have started a new chapter here to fill in the time gap with a summary of the latest research works. Based on the latest research developments, I will review my original research design as well as reflect on it. At the same time, I propose, based on the current literature review, how I would have designed the research and video captioning model.

2

Literature review

In order to solve the first research question, we start with a literature review to investigate the contemporary state-of-the-art of video captioning and applications for the visually impaired in the related field.
In this literature review, firstly, we give a brief introduction in section 2.1 on video caption as well as an explanation of some terminologies used in the field. Next, we dive into the pool of different video caption methods from the perspective of template-based generation, seq2seq model, reinforcement learning, and adversarial learning. In section 2.3, we present relevant video datasets in the area of video understanding and highlight ones that are significant to the video captioning task. Later, existing evaluation and readability metrics are reviewed and further discussed in terms of their merits and drawbacks in sections 2.4 and 2.5. In the coming section 2.6, we show the practical application for the visually impaired in the field of visual understanding. In the end, we conclude this literature review and reveal some insights into it.

2.1. Introduction

Video captioning is the process of generating a caption for a sequence of video frames. A caption is a textual description of what happens in the given video. Broadly, video captioning can be split into two main parts: understanding the visual content and then describing it grammatically. Thus, it bridges computer vision(CV) and natural language processing(NLP) techniques together. With the advance of computer vision and natural language processing, image captioning (the process of generating a caption for a single image), as the fusion of two tasks, emerged first and obtained great success. Further, the fusion is moved to be on video.

Compared to well known image captioning, video captioning is different and even more challenging. Unlike image, video contains spatial, temporal and even audial information. Accordingly, it is more difficult to process. Video is a sequence of images. There is a dependency between them that need to be learned to comprehend its content. As video has sequential frames, it is easier to detect motion. However, not all objects and actions in the video are useful for caption generation.

Video captioning can be divided into three groups according to the characteristics of the output.

- **video caption** often refers to a task that involves automatic sentence generation to describe the main event/activity that happens in a short video clip. Fig. 2.2 gives an example.
- **video paragraph caption** refers to a task that generates multiple sentences or a paragraph for a short video.[177]
- **dense video caption** compared to the aforementioned two tasks, is a new emerging area. It aims to detect and describe multiple events in a video based on various timespan. The dataset used for this task is *ActivityNet Captions dataset*[62], which is shown in Fig. 2.1.

Although the concept of video captioning still needs to overcome many challenges to get an ultimate form, It is not hard to imagine its applications in real life, from alleviating social problems to overcoming technical puzzles:

- Helping Visually Impaired people. People with vision challenge need a navigation system that help them to live. The very first step of a navigation system is to understand the user's surroundings, where video captioning can help. Video captioning model can describe the surroundings in natural language. What's more, the generated can be easily turn into audio for the visually Impaired people.

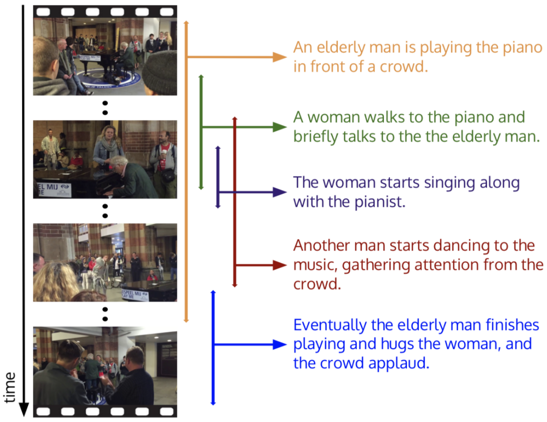

An elderly man is playing the piano in front of a crowd.
A woman walks to the piano and briefly talks to the the elderly man.
The woman starts singing along with the pianist.
Another man starts dancing to the music, gathering attention from the crowd.
Eventually the elderly man finishes playing and hugs the woman, and the crowd applaud.
time

Figure 2.1: ActivityNet Captions dataset[62] for dense video caption task.

A car drives along a track.

Figure 2.2: MSR Video-to-Text dataset[171] for video caption task.

- Helping people with speech impairment. Video captioning model can make the interaction with the speech impaired people easier. The model can be designed to understand the sign language(hand gesture and pose) and then translate it into natural language.
- Human Robot Interaction. It is no wonder that video captioning will pay a key role in human robot interaction(HRI). The video captioning technique can help the AI agent understand its surroundings and human instruction and then start a meaningful conversation with human.
- Storage Minimization. Videos consume many space to be stored in the computer, while text files occupy far less space. Therefore, transforming visual content into text data will save a lot of storage space.
- Video Surveillance. Suspicious activity detection in surveillance videos takes a lot time by naked eye. With the aid of video captioning, the tedious process can be automated.

2.2. Video captioning method

The development of video captioning methods is highly correlated to the surge of other fields of computer science, such as object detection, machine translation, deep learning, etc. In this part, video captioning methods are categorized and presented based on their backbone technologies individually. In each subsection, each classical method together with its posterior works is introduced. Furthermore, the advantages and limitation of these approach are discussed as well.

2.2.1. Template-based captioning

In early stage, researches on video captioning are mostly use language template to generate a sentence for the input video, hence here we call it template-based captioning. Generally, this method consists of two stages, content identification and sentence generation.

- **Content identification** is a stage where things like objects and activities are detected from the visual content. Thanks to the success of object detection and activity recognition, techniques from these two areas consecutively contributed to this stage. For instance, to detect an object in the video, HAAR features[159] or Scale Invariant Feature Transform(SIFT)[92] feature matching is used; to recognize activity, models like dynamic Bayesian Networks, Hidden Markov Models have been employed.
- **Sentence generation** is focused on selecting the detected words from the former stage and filling them into a language template for grammatically correct sentences. The template here refers to a predefined sentence structure with blanks for certain categories of syntactic components(like grammatical subjects, verbs, objects, etc.). The detected words from the first stage serve as lexicons, and only the most suitable ones are picked to form the sentence.

The earliest research in this line of work is [61] in 2002, which proposes an activity concept hierarchy to describe the action of a single person in the video. Given that this work highly relies on the correctness of activity concept hierarchy and the videos used are in constrained environments, it can barely be used in different situations. After that, more works appeared in this field, while most of them still tackled videos in restricted domains and small vocabulary corpora.

In 2013, [63] started to work on open-domain YouTube videos(This dataset consists of 1,967 short YouTube video clips whose duration is 10 to 25 seconds, and each of them contains a single activity) while the subjects and objects used are still restricted to 20 entities. Their main contribution lies in using extra text corpora to help screen SVO tuples (subject-verb-object tuples). Inspired by [63] and [41], [147] utilities, a comparatively larger number of entities and activities (45 candidate entities for the grammatical subject, 241 for the grammatical object, 218 candidate activities for the grammatical verb). Diffident from ancestors, this work involves a scene detector which can recognise a list of 12 scences/places in the video.[1] Besides, the Factor Graph Model(FGM) is proposed to aid in selecting optimal SVOP tuples (subject-verb-object-place tuples) by using language statistic from four extra text corpora.

To get rid of template-based engineering methods to generate captions, [130] treats video captioning as a machine translation problem. Firstly, they learn a CRF model to get an intermediate semantic representation of the input video. Then the semantic representation is translated to a natural language sentence via phrase-based statistical machine translation. Consequently, the semantic representation serves as the source language and the generated sentence as the target language. For training, they explore a parallel corpus of videos and textual descriptions and evaluate their work on *TACoS dataset*[123] (which annotates the cooking activities in the indoor environment and is discussed further in 2.3).

In summary, the use of language templates, although it ensures the grammatical correctness and completeness of synthesized sentences, leads to poor description in terms of sentence syntax. Besides, this method, as well as the statistical method, may face a great challenge to scale to open domain videos like MSR-VTT dataset[171] (which is discussed further in section 2.3) because a simple template is less flexible and insufficient for large video dataset that contains an unforeseeable number of subjects, objects, activities, and places. In other words, a manual-designed template is costly and even infeasible in such a case. Additionally, The separation of two stages ignores the interplay between the visual content and linguistic pattern. Accordingly, the joint latent space between visual and linguistic representations is not taken into consideration as well.

[1] mountain, pool, beach, road, kitchen, field, snow, forest, house, stage, track, and sky

2.2.2. Deep learning methods

Thanks to the availability of GPUs and CPU clusters as well as large amounts of training data, recent deep learning brings great success to computer vision (CV) and also gradually leads to good performance on natural language processing (NLP) tasks like machine translation, etc. Video caption, such a task that bridges CV and NLP together, has been revolutionized by deep learning as well, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs excel at analyzing visual imagery, such as object recognition, while RNNs have great influence over sequence modeling in machine translation and image captioning. In the area of video captioning, the research community has integrated these deep neural networks in various forms and has had an encouraging performance.

Compared to template-based methods, deep learning methods are data-hungry, which allows video captioning to extend from fine-grained to open domain. In addition, deep learning relieves the burden of manually crafted features and manually designed language templates by multiple-layer neural network learning. In the following sections, the mainstreaming Encoder-Decoder framework of deep learning methods is introduced firstly in section 2.2.2.1 and how other techniques like attention mechanism, deep reinforcement learning, generative adversarial networks, and transformer improve this framework are demonstrated in section 2.2.2.2, 2.2.2.3, 2.2.2.4, and 9.2.2 sequentially.

Encoder-Decoder

Motivated by the work in machine translation and image captioning, some early works treat video captioning as a machine translation task and propose an Encoder-Decoder structure to solve it. Similar to the template-based method, this method can be divided into two stages sequentially as well:

- **Encoding stage** is a stage where the CNN, RNN, LSTM, or any other DNN is employed to encode video into semantic representation and then feed them to the next stage.
- **Decoding stage** is also known as text generation, in which stag, the semantic video representation from the forming stage, is translated to natural language sentences. Different variants of RNNs, such as deep RNN, bi-directional RNN, LSTM, or GRU, are used for decoding.

[33] presented the LRCN model, the first application that employs deep learning in the video captioning task. Have seen the overwhelming performance of LSTM shown in the machine translation task, they replaced the statistical machine translation in [130] (which has been discussed in section 2.2.1) with the same two-layer LSTM for encoding and decoding. Different from the LRCN model[33], where CRF plays a role in getting supervised intermediate representation, [156] was proposed in end-to-end fashion by directly appending the LSTM to a deep CNN. They adopted a pretrained CNN to generate a fixed-length feature vector from the input video to avoid manual feature selection. To be specific, They implemented a mean pooling of over-extracted features across all sampled frames and generated a single one-dimensional vector to represent the entire video. This actually reduces the video captioning task into an image captioning one. Then the feature vector is fed into a two-layered LSTM as input at every time step, as the result, the caption is generated. Although mean pooling is a simple and cheap way to represent the video, in later literature [157], it has been pointed out to collapse the temporal structure of a video.

Later, the combination of CNN and LSTM architecture became the mainstream form for video captioning tasks. Many works extended it by learning a joint embedding space between video and sentence[108], adding extra semantic attributes(such objects attributes from frames and actions label from video flow)[127], adding extra knowledge to decoder[158], etc. Although the Encoder-Decoder framework makes video captioning in the open domain possible and most later works are based on it, early models either ignore the temporal structure of video by simply reducing video captioning to image captioning or insufficient utility temporal information of the video.

Attention Mechanism

Nowadays, Attention Mechanism is one of the most influential concepts in the Deep Learning community. Inspired by humans that allocate consideration unevenly (focus on things that they are interested in while ignoring or diminishing the importance of others), Attention Mechanism learns to make choices about which features they pay attention to. Initially, it was designed for machine translation to adaptively control how much each hidden unit remembers or forgets while reading/generating a sequences in Seq2Seq Model.[24] Later, when integrated with neural word embeddings, Attention Mechanism becomes a crucial component of algorithms like Transformer [154] and BERT [30] that is quite a leap in setting new benchmarks in Natural Language Processing tasks. In addition, the attention mechanism has been widely used in visual captioning

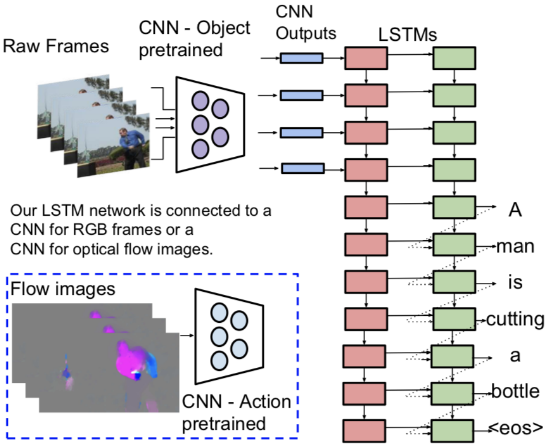


Figure 2.3: The illustration of S2VT. [157]

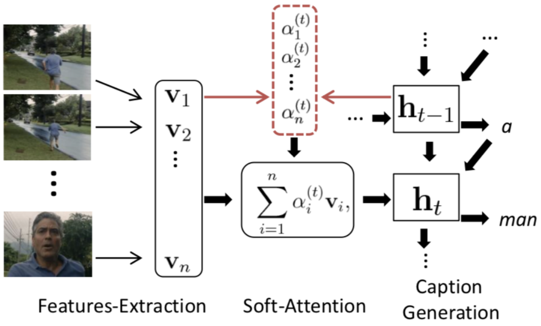


Figure 2.4: The illustration of temporal attention. [175]

tasks. The task of image captioning mainly explores spatial attention mechanisms. Take [172] as an example, they employed visual attention mechanism to automatically focus on regions-of-interest (ROI) in the images and then generate a word according to it. In contrast, the utility of the attention mechanism in video caption tasks is comparatively complex since video is a multi-modal carrier.

To exploit the temporal dynamics of the video further, [175] proposes an approach for exploiting both the local and global temporal structure. In order to capture local motion dynamics, they used a 3D convolutional neural network(3D CNN) [151]. The 3D CNN is pre-trained on an activity recognition dataset so as to produce a higher-level representation that is tuned to human actions from short-frame sequences. This is achieved by first dividing the input video clip into a 3-D spatio-temporal cuboid, and each cuboid is represented by concatenating the histograms of oriented gradients, oriented flow, and motion boundary. For the global temporal structure, they incorporated the soft attention mechanism with the decoder to learn the long-term dependencies and ordering of activities. As shown in Fig. 2.4, the soft attention mechanism generates a vector of attention weights α_n^t for all selected frames at each time step t based on the previous hidden state h_{t-1} from the decoder (which presumably summarizes all the previously generated words) and the corresponding frame's temporal feature vector V_n. Instead of a simple averaging strategy used in [33], the dynamic weighted sum of the temporal feature vectors according to attention weights generated at each time step is fed into the caption generator (the decoder). The attention mechanism makes the decoder capable of focusing on a certain subset of frames by increasing the attention weights $alpha$ of the corresponding temporal feature V and vice versa. In later work for video captioning, adopting temporal attention in the decoder has become very popular.

Although temporal attention helps to capture the global temporal structure of videos, it does not pay attention to multiple salient objects within a single frame, which may lead to detail missing in captions. Inspired by the aforementioned work [172], [152] proposes a spatial-temporal attention (STAT) method for video captioning to handle this issue, which can selectively focus on a certain subset of frames as well as salient objects in that subset. As illustrated in Fig. 2.5, STAT consists of three layers. Layer 1 is the spatial attention where different weights are assigned to local features (the semantic embedding of objects detected by faster-RCNN) for each frame. Layer 2, the temporal attention, works separately on global-motion features (the concatenation of frame-level features extracted by GoogLeNet and motion dynamics captured by C3D) and the weighted local features from Layer 2. Then, Layer 3 fuses the two temporal representations from Layer 2 and forms the new representation for the decoder at each time step. This method benefits from incorporating the object attributes from frames into attention mechanism, thus it can generates relatively detailed and accurate captions.

Apart from selectively focusing on the frames and ROI like [152], [51] expands the attention model to selectively attend input features across modalities like visual features, motion features, and audio features, namely multimodal attention. Fig. 2.6 shows how multimodal attention works together with temporal attention: The multimodal attention stacks on top of two temporal attentions, which are applied to two features of different modalities, respectively. Multimodal attention weights are obtained in a similar way to the temporal attention mechanism.

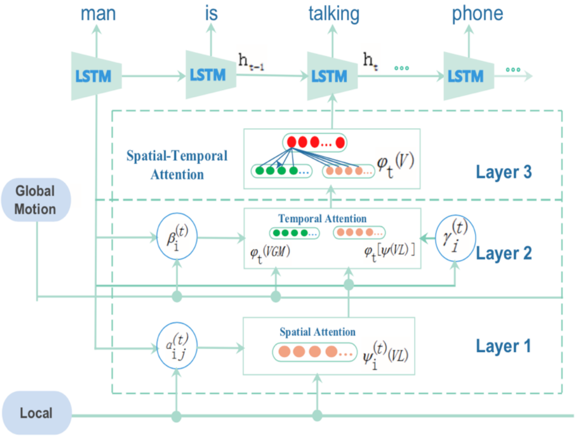


Figure 2.5: The illustration of spatial-temporal attention (STAT). [152]

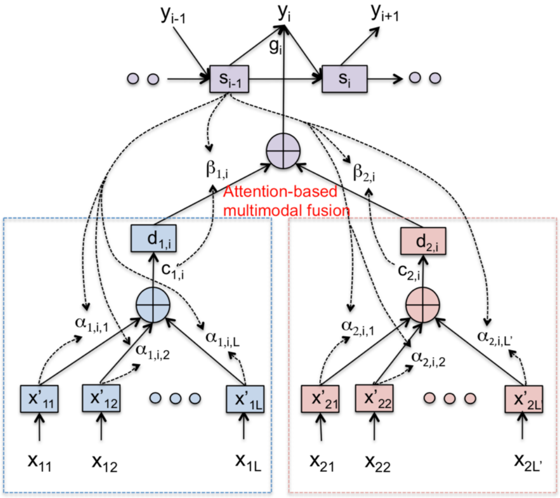


Figure 2.6: The illustration of multimodal attention mechanism. [51]

As the underlying concept of the attention mechanism is to mimic the human's way of thinking, much effort was put into introducing extra data from human perception to further improve the preciseness of the attention mechanism. In their work, Sun et al. add an extra step in the encoding phase, where multi-modal semantic attributes will be detected and generated for video. Then, this high-level semantic attribute is passed to the decoding phase to guide the attention mechanism to generate a better description. Human gaze data, as another type of human perception, is utilized by Yu et al. to find the Region of Interest (ROI) in the video frame. The model of Chen and Jiang proposed is reported that the spatial attention is better supervised by the motion learned from stacked optical flow image. Unlike other methods, SGN[133] aligns phrases with a collection of relevant frames, which are called semantic groups in this work. Contrastive attention loss is proposed to ensure the inner coherence of semantic groups with respect to meaning, especially when they are free from human annotation. In other words, the Contrastive Attention Loss takes advantage of human perception and guides the model to generate better captions.

Most recently, Transformer[154] has attracted tremendous interest across multiple disciplines. In the principle, Transformer can be regards as "a stack of attention layers" but it still follows the encoder-decoder structure. The details of the Transformer are discussed in section 9.2.2.

Deep reinforcement learning

Reinforcement learning (RL) is generally an artificial intelligent agent for sequential decision-making problems, which learns an optimal policy from rewards or punishments of the environment by trial and error [140]. Owing to powerful computation and big data, the integration of reinforcement learning and deep learning, aka 'deep reinforcement learning' (DRL), has achieved great success in many real-world games against humans. For example, AlphaGO[135] makes remarkable achievements in the two-player zero-sum game, and its descendant AlphaGO Zero[136] also makes great progress in chess and Shogi. Recently, the research community has raised interest in using the DRL technique to tackle the problem of captioning tasks as well.

Standard video captioning models (most of the aforementioned models) typically minimize cross-entropy loss during training while their performance is assessed on other discrete metrics such as CIDEr[155], METEOR[8], etc. (Metrics are further discussed in section 2.2). This is known as the objective mismatch. Nevertheless, these models suffer from exposure bias as well. During training, these models are exposed to ground truth during training. During testing, models rely on the former predictions to generate the next output. If the previous prediction is wrong, the error will accumulate and ultimately lead to terrible results. The most straightforward way to fix the first issue is to optimize the same metric both in training and testing. Meanwhile, the commonly used metrics are not differentiable. For the second issue, scheduled sampling may help since it gradually exposes models to input words from the model distribution instead of the ground truth. However, it might prevent diverse caption generation to a certain degree [111].

To tackle with aforementioned two problems, [121] propose a Mixed Incremental Cross-Entropy Reinforce (MIXER), whose underlying concept is to use hybrid loss function to optimize the score of BLEU[109] metric in the end. Initially, the model utilizes cross-entropy loss for pretraining, then slowly deviates from using reinforcement learning with a baseline for finetuning where sentence generation relies on its own predictions, as is done at test time. After that, the BLEU score is used to compute the reward and update the sentence generator. The later work [124] proposes self-critical sequence training (SCST) for image captioning, which basically inherits the MIXER structure from [121] but employs the output of the greedy decoding as the baseline at each time to avoid high variance. Another empirical finding in this work is when optimizing the CIDEr[155] metric, the scores of all other metrics increase. Since then, most of the later work has used CIDEr or its variants as the reward. Based on the previous insight, [113] propose a Self-Consensus Baseline (SCB) for video captioning. Different from SCST, using greedy decoding as a baseline, they calculate the mean reward of all ground-truth sentences of the same video and allow training on several captions at the same time. Using frame-level features simply concatenated with C3D, MFCC(as the audio feature), and category information, this model achieves the best performance against the popular benchmarks from our knowledge.

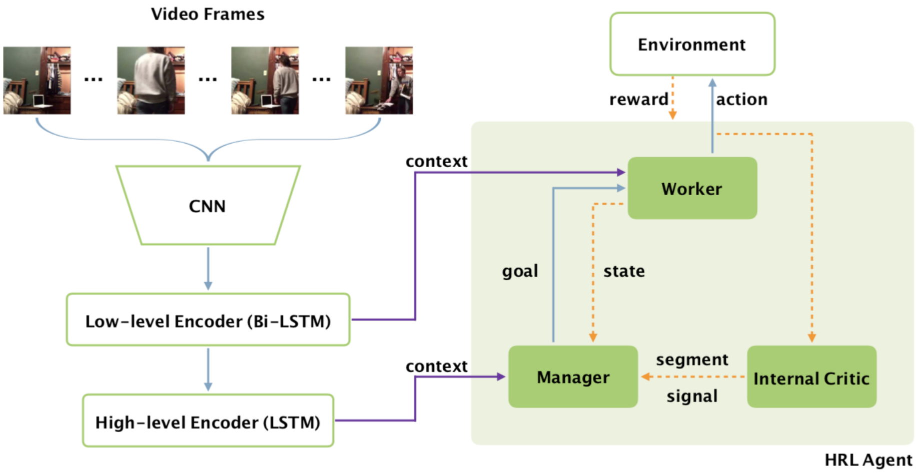


Figure 2.7: The illustration of HRL video captioning.[162]

In order to grasp the semantic flow of a higher level, [162] proposes a fully-differentiable neural network with hierarchical reinforcement learning(HRL) for video captioning, as shown in Fig. 2.7. This work still follows the encoder-decoder architecture. A pretrained CNN, Bi-LSTM, and LSTM are employed in the encoding stage. At the same time, the HRL agent serves as the decoder, which has three components: the high-level Manager, the low-level Worker, and an internal critic. The underlying concept is "Divide and conquer": The Manager generates the context between textual segments, and the Worker follows the guidance of the Manager to generate segments sequentially. The internal critic decides whether the worker completes segment generation or not. With such a compositional and complex framework, this work does not significantly outperform its baseline methods from the perspective that its goal is to generate longer sentences.

From a different angle, [20] sheds light on frame selection in video captioning. In order to avoid redundant visual information and expensive computation costs, they propose *PickNet* to select the most informative set of frames by rewarding visual diversity and penalizing textual discrepancy. It was presented to be a design that makes a trade between speed and accuracy. From their experiments, Using 6 to 8 frames can get competitive results against contemporary works.

In the most recent, [76] introduces Multitask Reinforcement Learning to video captioning, learning a video representation and a reward together at the same time to regulate the search space for caption generation. Similar to MIXER[121], they pretrain the model by cross entropy loss and reinforcement learning reward consecutively in stages 1 and 2. In stage 3, inspired by [169], they provide extra gradients from a video attribute prediction module directly to the encoder and jointly train the encoder and decoder by minimizing the convex combination of the attribute loss and the REINFORCE loss. In contrast to previous work that uses CNN only as the encoder to extract features, it works in a truly end-to-end learning fashion. Although it does not considerably outperform other RL-based work, this work shows that the domain-adopted video representation(the predicted video attributes) is more powerful than the generic frame-level features from its ablation studies.

To conclude, introducing RL to directly optimize a favored metric in the training stage addresses the exposure bias and objective inconsistency. It definitely achieves a higher score on the video captioning benchmark than previous work. However, there is no theoretical proof that optimizing one metric can gain consistent improvements in overall metrics, and the automatic metrics still fail in low correlation with human judgment. Besides, RL-based methods have been reported to have unwanted artifacts sometimes, like ungrammatical sentence endings [44], higher object hallucination [129], and lack of diverse content [32]. Taken above issues into consideration, the research community start to integrate RL with adversarial learning in visual captioning. Work in this line is discussed in next section.

Generative Adversarial Network

Regardless of unsupervised learning or semi-supervised learning, Generative Adversarial Networks (GANs)[40] provide us with a new way of thinking about problems, which is to introduce game theory into the machine learning process. The original GAN, proposed by Goodfellow et al.[40] in 2014, consists of a generative model and discriminative model: the generative model tries to capture the data distribution and produces similar data while the discriminative model learns to tell whether the data is from data distribution or the model distribution. In other words, the generative model is trained to fool the discriminative model, generating data as realistic as possible that the discriminative model can not distinguish. After the introduction of the GAN, many variants have been proposed to avoid instability and mode collapse [101][116][5][42] or reformulate it from different perspective[18]. One worth mentioning is the Conditional Generative Adversarial Network (CGAN)[101], an earlier improved version of the original GAN. The original GAN does not have control over what to generate since the output only relies on random noise. To handle this problem, CGAN adds a conditional input apart from random noise to the model, providing extra information to guide the data generation process. The conditional input here, in theory, can be anything related to the output, like the image label, attributes of the object, or text embedding. Although the modifications are really simple, the idea is very instructive.

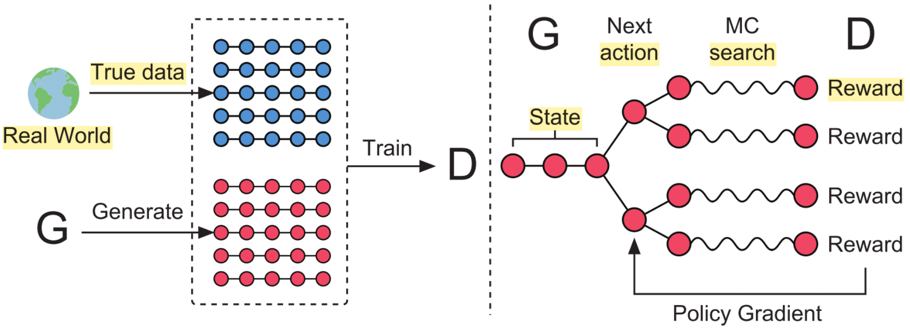


Figure 2.8: In the Left, the real data and the generated data from Generator (G) are used to fool the Discriminator(D). In the right, G directly uses policy gradient to update its state and decide its next action, in which the final reward is from D and calculated via Monte Carlo search.[178]

Recently, GANs have been adopted by computer vision(CV) and achieved great success in tasks such as image synthesis[67][122], image transfer[187], etc. Research on applying GAN in natural language processing(NLP) is also an intriguing trend, where training GAN is tougher since many NLP tasks are discrete domain generation issues (GAN was originally designed for continuous data). Although GAN makes encouraging achievements in real-value data synthesis, the difficulty in differentiation for discrete data and evaluation of partially generated sequences hinder conventional GAN from succeeding in NLP tasks like text generation. Inspired by the idea of reinforcement learning, SeqGAN[178] treats text generation as a sequence decision-making problem and models the generator as a stochastic policy as shown in fig.2.8. To solve the differentiation problem, the generator directly uses a policy gradient to update its parameters. Monte Carlo Tree Search (MCTS) with a roll-out policy is employed to sample the unknown remaining tokens and approximates the state-action value to evaluate the partially generated sequence at an intermediate step.

Both CGAN and SeqGAN have inspired work that integrates GAN into visual captioning models, as visual captioning tasks can be regarded as text generation tasks conditioned on the image/video content. On top of CGAN, [28] is the first work that introduces GAN to the image captioning task. To generate captions that are more diverse and less distinguishable from human ones, they trained a generator and an evaluator together and applied Policy Gradient with early feedback as well. It is reported to outperform the state-of-the-art in terms of naturalness, diversity, and relevance. From a different starting point, [176] adopts GAN architecture to train a binary classifier to be an effective metric. The binary classifier, which, in fact, is the discriminative model of GAN, acts as a human critique to tell human-written and machine-generated captions apart. Along with the lines of [28], the most recent work [12] proposes a similar framework, extending RL-based encoder-decoder architecture with CGAN and SeqGAN. In contrast to [28], this work makes a trade-off between naturalness and fidelity (achieve improvements in overall evaluation metrics). Another latest work in this line is [74], which generates diverse captions across images. They propose a comparative adversarial learning framework, where the discriminator evaluates the quality of captions by comparing them with other captions within the image-caption joint space.

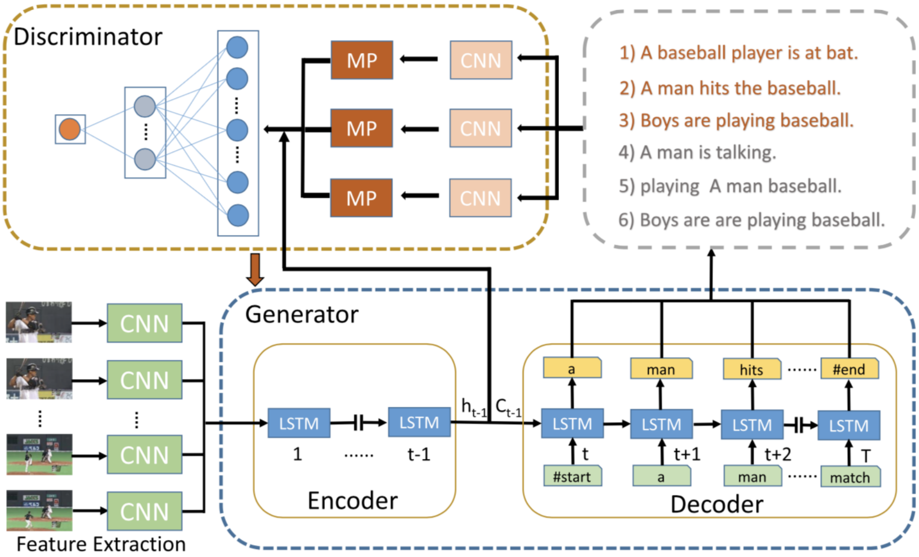


Figure 2.9: The generator aims to generate a sentence for the video as relevant as possible, but the discriminator is designed to tell the generated sentences from reference sentences. The orange sentences that feed into the discriminator are the reference sentences, while the black sentences are synthesized by the generator. Otherwise, badly constructed sentences or uncorrelated sentences are generated by the generative model. MP here represents the max pooling layer. [173]

LSTM-GAN[173] is the first attempt that applies GAN to an LSTM-based video captioning task. Although LSTM is seen to be promising in the sequence generation task, it suffers from exponential grammatical error accumulation. This may lead to generating words of lower association when the video length increases. The authors[173] thought introducing GAN to distinguish if the generated sentences are relevant to the video can further compensate for the deficiencies of LSTM. Identical to other GAN models, LSTM-GAN contains a generator and discriminator as shown in figure 2.9: The generator aims to generate a sentence for the video as relevant as possible, but the discriminator is designed to tell the generated sentences from reference sentence. To enhance the classifying ability, the discriminator is designed to have a convolutional layer, a max-pooling layer, and a fully connected layer sequentially. Motivated by CGAN[101], the discriminator takes not only the sentences as input but also the video features from the Encoder of the generator as extra information. When it comes to the discrete data differentiation problem of GAN, rather than using reinforced learning like [176] [163], they use an embedding layer[52] to convert the discrete outputs into a continuous representation. As far as we known, LSTM-GAN marginally improve the quality and diversity of the generated sentence, meanwhile, it is potential to perform better by taking advantage of Reinforcement Learning.

I have seen the exploration of reinforcement and adversarial learning in image/video captioning tasks, and the research community has applied them to other branching tasks: visual storytelling [163], video paragraph (multi-sentence) captions [110]. [163] proposed an adversarial reward learning scheme for synthesizing abstract stories for photo streams, where a reward function is learned from the human description. Noticing the difficulties of jointly training the GAN model(If the discriminator is too strong that the generator gradient vanishes and learns nearly nothing.[4]), [110] trained the discriminator to pick up the best one from a collection of sampled candidate sentences during the inference of generator, namely Adversarial Inference. Besides, they used the three discriminators for sentences: semantic, relevance to visual features, and cohesion across sentences, respectively. This model has been reported to outperform previous approaches like

SCST and GAN in human evaluation.

In summary, Applying adversarial training to visual captioning tasks encourages more diverse and "human-like" captions as opposed to optimizing a metric. GAN loss is shown to be helpful to decrease object hallucination as well[129]. Although some trials on integrating GAN together with RL to image captioning model have been witnessed, only a few works on applying them to video captioning model have been found so far, probably due to the complex nature of video data. Accordingly, till now, no significant improvement have been reported. In addition, GAN has its own limitations. It is difficult to achieve stable training because of the discrete output space and its lack of coherence in text generation. Still, the concept of learning to tell human-written sentences from the machined-generated ones is very compelling and promising.

2.2.3. Discussion

Currently, the interest of the research community in the field of video caption has shifted from a template-based method to a deep-learning method. Especially, recent years have witnessed the distinguishing performance that transformers achieved in machine translation tasks. Transformers become popular across different disciplines. Research work related to transformers in video captioning tasks started to emerge. In this section, the merits and drawbacks of different methods are concluded.

Template-based captioning

The use of language templates ensures the grammatical correctness and completeness of synthesized sentences at the cost of poor description in terms of sentence syntax. Besides, the majority of template-based methods isaretatistical, which leads to poor scalability to open domain videos. A simple template is less flexible and insufficient for a large video dataset that contains an unforeseeable number of subjects, objects, activities, and places. In other words, a manual-designed template is costly and even infeasible in such a case. Additionally, The separation of two stages(encoding and decoding) ignores the interplay between the visual content and linguistic pattern. Accordingly, the joint latent space between visual and linguistic representations is not taken into consideration as well.

Deep learning method

Compared to template-based methods, deep learning methods are data-hungry, which allows video captioning to extend from fine-grained to open domain. In addition, deep learning relieves the burden of manually crafted features and manually designed language templates by multiple-layer neural network learning. In the following paragraphs, the pros and cons of various deep learning methods applied in the video captioning task are discussed.

- Attention mechanism. The attention mechanism usually goes with the encoder-decoder framework. In essence, the attention mechanism enables each output from the decoder to access the entire input sequence by retaining each hidden state of the input sequence in the course of decoding. This means the decoder can choose specific elements from that input sequence to produce the output.[96] As noticed, temporal attention is good at capturing global information across multiple frames but fails to interpret the details of specific frames. That is exactly why later spatial attention is proposed to find regions of significance. There are many variants of attention based on these two types, generally it can be classified into two directions: attention for modality selection and guidance for attention. Since the attention mechanism more or less is an add-on component of the encoder-decoder framework, it faces the problem of recurrent networks like the inability to scale up and parallelize.
- Reinforcement learning. Introducing RL to directly optimize a favored metric in the training stage is to address the exposure bias and objective inconsistency. It definitely achieves a higher score on the video captioning benchmark than previous work. However, there is no theoretical proof that optimizing one metric can gain consistent improvements in overall metrics, and the automatic metrics still fail in low correlation with human judgment. Besides, RL-based methods have been reported to have unwanted artifacts sometimes, like ungrammatical sentence endings [44], higher object hallucination [129], and lack of diverse content [32]. Taking the above issues into consideration, the research community started to integrate RL with adversarial learning in visual captioning.
- Adversarial training. applying adversarial training to visual captioning tasks encourages more diverse and "human-like" captions as opposed to optimizing a metric. GAN loss is shown to be helpful to decrease object hallucination as well[129]. Although some trials on integrating GAN together with RL to image captioning models have been witnessed, only a few works on applying them to video captioning

models have been found so far, probably due to the complex nature of video data. In addition, GAN has its own limitations. It is difficult to achieve stable training because of the discrete output space and its lack of coherence in text generation. Still, the concept of learning to tell human-written sentences from the machined-generated ones is very compelling and promising.

2.3. Dataset

The prevalent cluster of well-annotated datasets is highly associated with the surge in the research area of video caption, which has been concluded in Table 2.1.

Table 2.1: Dataset used for video captioning

Dataset	Domain	Caption
KTH dataset[134]	six human action	single sentence
UCF101[138]	sport-related YouTube videos	single sentence
Sports1M[56]	sport-related YouTube videos	single sentence
HMDB_51[64]	movie clips	single sentence
Hollywood2[97]	movie clips	single sentence
MPII MD [128]	movie clips	single sentence
M-VAD[148]	movie clips	single sentence
TaCoS Dataset[123]	cooking	single sentences
TACoS-Multi Dataset[126]	cooking	multiple sntences
MSVD dataset[13]	open domain	single sentences
MSR Video-to-Text [171]	open domain	single sentence
Activitynet[38]	human activities	multiple sentences
Object-oriented captions[186]	human activities	multiple sentences
UET Video Surveillance (UETVS)[31]	Surveillance	sentences

Earlier, datasets for video understanding are mostly narrow in the degree of visual content and simple in terms of annotations. The first video database, KTH dataset[134], is centric on six human actions [2] in grayscale. After that, several datasets throw some light on human action recognition in sport-related YouTube videos, such as UCF101[138], Sports1M[56] and etc. Then HMDB_51[64] and Hollywood2[97] lead video understanding to human action in movie clips. In the cooking domain, TaCoS Dataset[123] annotates the cooking activities in the indoor environment with sentences.

However, the Microsoft Video Description MSVD dataset[13], which is also called YouTube2Text, is one of the earliest datasets in the open domain and is still widely used today. It consists of 1,970 video clips with 80,839 sentences in total, covering a number of topics like animals, sports, food, etc. Each clip lasts for around 10 seconds and is annotated with approximately 41 parallel sentences ('parallel' here means the sentences are from different Amazon Mechanical Turk workers). Every sentence contains about 8 words and is available in various languages. As a result, MSVD has a vocabulary of 13010 unique words.

Later, with a zeal for bridging video semantics to language, more datasets are released. The extension of the aforementioned TaCoS Dataset, TACoS-Multi Dataset[126], uses paragraphs for annotation but still narrows in the cooking scene. MPII MD [128] and M-VAD[148] are two new large-scale movie description datasets. Both of them employed Descriptive Video Service(DVS) narrations (DVS is a service that makes multimedia, especially visual content, accessible to people who are blind or visually impaired.)[3] and the annotation of the latter one is manually aligned with movie clips.

Recently, MSR Video-to-Text [171], another open-domain dataset, was released and soon became one of the most popular video captioning benchmarks due to its largest amount of clip-sentence pairs (200K in total). Similar to the MSVD dataset, each clip in MSR-VTT is annotated with multiple independent sentences as well (20 sentences per clip). All 7180 videos in this dataset are collected from the online search engine, covering 20 representative categories[4].Those videos are segmented into clips of 10 to 30 seconds. Altogether, it is 41.2 hours long and contains 29,316 unique words. Besides, this dataset encloses an audio channel, which may be used as the multimodal feature.

In the recently dense video captioning task, Activitynet[38] captions dataset is collected in order to handle multiple (time-overlapping) events in a video. It contains 20,000 open-domain videos that are 180s long on average. Different from MSVD and MSR-VTT, every video is captioned with a paragraph, and each event that occurs in the video is temporally localized and described in one single sentence. And sentences for their

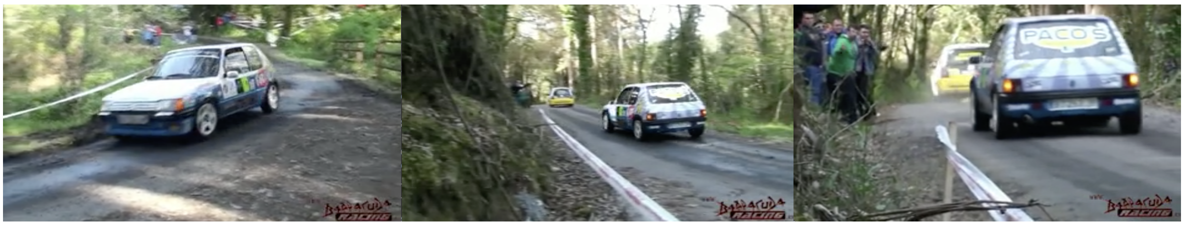
[2] walking, jogging, running, boxing, hand waving, and hand clapping

[3] http://main.wgbh.org/wgbh/pages/mag/description.html

[4] These 20 categories include music, people, gaming, sports (actions), news (eventspolitics), education, TV shows, movie, animation, vehicles, how-to, travel, science (technology), animal, kids (family), documentary, food, cooking, beauty (fashion), advertisement.

respective videos retain the contextual information between events. Every video is captioned by an average of 3.65 sentences with an average length of 13.48 words, which leads to a total of 100,000 sentences in this dataset. Remarkably, Activitynet is an action-centric dataset involving a large portion of human activities.
Based on ActivityNet[38], Zhu et al. construct a new dataset Object-oriented captions[186]. They select videos from 'playing games' class of ActivityNet and re-annotate them by explicit object-sentence pair. They claim this dataset is more diverse in terms of activities and scenes than other datasets. Unfortunately, object-oriented captions are not allowed to be downloaded.
UET Video Surveillance (UETVS)[31], as its name implies, is a domain-specific dataset used for surveillance purposes. It consists of 1200 videos. Each video is annotated with 3-6 parallel sentences. The dataset is not available for download.

Since the scope of our work is focus on video captioning in the open domain for people who are blind or visually impaired and the video is limited to 30s , we will use MSR-VTT dataset in this work. The detail of dataset analysis will be given in the following chapter.



C1: a car drives along a track
C2: racers drive around the corner
C3: a group of people watching cars race
C4: cars are racing down a mountain path
C5: an old car with the logo packs is racing against another car
C6: cars are traveling down a road surrounded by people in a forest

Figure 2.10: an example of MSR-VTT dataset

2.4. Evaluation metrics

The evaluation of the video caption is a considerably subjective task since a video can be validly described in different ways; in other words, there is no specific ground truth for evaluation. As the example taken from the MSR-VTT dataset shown in Fig. 2.10, several given sentences all correctly describe the video clips but vary in syntax, semantics, and attention. For instance, Sentence C5 is more focused on the details of the racing car ("an old car with logo packs"), while Sentence C4 gives more information about the scene ("a mountain path"). Nowadays, both human evaluation and automatic evaluation are employed to determine the model's performance in this task.

2.4.1. Human evaluation

Human evaluation, just as its name implies, is letting people manually judge the quality of the generated sentences. It can be implemented by using crowdsourcing tools such as AMT(*Amazon Machine Turk*[5])or expert knowledge. Generally, machine-generated sentences can be evaluated from two aspects: correct grammar and relevance. In items of relevance, sentences are ranked, and only two sentences that are exactly the same will have the sink scorerank. Despite the subjectivity, human-based evaluations are time-consuming and labor-costly as well. Thus, the research community tends to use automatic evaluation metrics as alternatives to develop models and compare them with their works. These metrics calculate a single score, which represents the similarity or dissimilarity between the model-generated sentence and a set of reference (human-annotated) sentences.

2.4.2. Automatic evaluation

When it comes to automatic evaluation for video captioning, commonly used metrics are mainly borrowed from machine translation and image caption tasks, which are summarised in Table. 2.2. The majority of

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

metrics(BLEU, ROUGE, METEOR, and CIDEr) are based on lexical similarity, counting matches between n-grams of the generated and the reference captions by applying different formulas. However, SPICE measures the semantic similarity of candidate and reference captions by using *Scene Graphs*, and WMD calculates the Word Mover's Distance between them by utilizing the *word2vec* embedding. In the following several sections, we are going to discuss the strengths and weaknesses of those metrics further.

Table 2.2: Automatic Metrics used for video caption evaluation

Metric	Original Task	Main concept
BLEU[109]	Machine translation	n-gram precision
ROUGE[82]	Document summarisation	n-gram recall
METEOR[8]	Machine translation	n-gram with WordNet
CIDEr[155]	Image captioning	tf-idf weighted n-gram
SPICE[2]	Image captioning	Scence-graph
WMD[66]	Document similarity	Word Mover Distance on *word2vec*

BLEU

BLEU (Bilingual Evaluation Understudy[109]) is proposed as a quality measure of machine translation, measuring the similarity between the machine-generated text and that of a human. It is defined as the sum of the geometric mean of n-gram precisions and a simple term that penalizes the generated text that is shorter than the reference ones. Specifically, in the video captioning task, BLEU measures how well the correspondence between a generated sentence and its multiple reference sentences by calculating the rate of the coincident n-grams between them in the generated caption. From its definition, we can see that BLUE is simple and fast to compute. Meanwhile, BLUE is highly affected by the word choice, word order, and sentence length and ignores the structure and the semantics of the sentence. This may lead to unauthentic evaluation of sentences that are of less overlapping n-grams with the reference ones but are still semantically similar. Additionally, Using BLEU to evaluate a single sentence might not be fair because it is initially for assessing documents. In this thesis, the smoothed version of BLEU is used.

ROUGE_L

On the contrary to BLEU, ROUGE(Recall Oriented Understudy for Gisting Evaluation[82]) is a recall-based n-gram metric. ROUGE is first employed in document summarisation, and its variant ROUGE_L is popular in image and video captioning. This metric computes the F_1 score of the longest common subsequences(LCS) in each candidate-reference sentence pair. LCS is defined as the in-sequence matches, other than consecutive matches, which reflect sentence-level word order. Because it always seeks for the longest in-sequence common n-grams, we do not need to preset the n-gram length. However, ROUGE_L has bias as well. It prefers longer sentences due to its dramatic dependence on recall.

METEOR

METEOR(Metric for Evaluation of Translation with Explicit Ordering [8])is explicitly designed to deal with the drawbacks of BLEU by introducing recall and exact word matching. It is computed based on the harmonic mean of the precision and recall of uni-gram alignments between the generated and reference sentences, together with a penalty that involves longer n-gram matches. As aforementioned, BLEU does not cater to the semantic content of sentences. To some extent, METEOR solves this issue by integrating WordNet-based synonym matching, but this metric is still not capable of capturing sentence-level semantic similarity. In a study in 2014[37], METEOR is reported to be the best metric amongst BLEU and ROUGH in terms of correlation with human evaluation.

CIDEr

CIDEr(Consensus based Image Description Evaluation[155]) is a metric proposed specially for image captions, it extends the contemporary metircs with IF-IDF (term frequency and inverse document frequency) on n-grams of higher order. To be specific, it stems all words in generated and reference sentences and treats every sentence as a bag of n-grams(1 to 4 words). Later stemmed n-grams are encoded by IF-IDF. (Under the assumption that common words across the whole dataset are less informative in terms of visual content, the n-grams frequently occurring in the whole dataset will be lower weighted. In contrast, commonly appearing n-grams in the reference sentences will be weighted higher). In the end, CIDEr measures the mean cosine similarity between n-grams from generated and reference sentences. Theoretically, using higher order of IF-IDF weighted n-grams helps to get better grammatical properties and semantics. In fact, IF-IDF may also overweights some unimportant images details, which may lead to unfair evaluation.

SPICE

SPICES(Semantic Propositional Image Captioning Evaluation[2]) is another recently proposed metric for image captioning. SPICES computes the F1-score between *the scene graph tuples* of sentences. Through the dependency parse tree, texts are transformed into semantic scene graphs, where a set of tuples contains three tokens: the object, its attributes, and its relationship. In the research of[89], Liu et al. demonstrates that SPICE tends to regard sentences with repeating clauses as good captions. However, till now, not much research work has used SPICES as an evaluation metric. Another disadvantage is evident. SPICES relies on parsing. The quality of parsing may affect its performance on caption evaluation.

WMD

WMD stands for Word Mover's Distance, which measures the semantical distance between texts in *word2vec*[99] (semantically meaningful vector representations) embedding space. Earth Mover's Distance(EMD)[132] is used to compute the distance between texts, which was originally employed to calculate the transportation cost. Two sentences share lots of the same words, but that does not mean they are semantically similar. Vice versa, two sentences that vary in word choice may still have the same semantic meanings. WMD is proposed to handle this case. As reported in [59], Compared to BLUE, ROUGE, and CIDEr, WMD is less sensitive to word order or synonym swapping. Further, similar to CIDEr and METEOR, it gives high correlation against human judgements.[59]

2.4.3. Conclusion

To summarise, automatic evaluation may face the risk of bias caused by the subjectivity of reference sentences annotated by humans. To treat this issue, introducing more reference captions is reported as an effective way to bring more reliable evaluation in [155]. Besides, each metric has its obvious merits and blind spots. In many works like [155] [37], CIDEr and METEOR are found to be more closely correlated to human judgments. Meanwhile, In study [59], CIDEr and SPICE are found to be much more sensitive to synonym replacement in sentences. BLEU, ROUGE, and CIDEr are highly affected by word order. Although SPICE and WMD are the two latest metrics that capture better semantics compared to others, SPICE ignores the fluency of sentences, and WMD is rarely reported in the literature at the time of this thesis.

2.5. Readability metrics

When it comes to readability, there are plenty of metrics that can be used. Some research implies that there is no significant difference among results across different readability metrics.[70] However, there are also some study indicates variance among results from different readability metric instead of the similarity.[165] Besides, there is some literature that gives evidence that different readability metrics may be suitable for different tasks. Considering our target audience is teenagers with visual impairment, we decided to use the following four readability metrics in the experiment. The details of these four metrics and the reason why we chose them are below.

The SMOG Index[69] is a readability metric where SMOG stands for 'Simple Measure of Gobbledygook'. It counts the number of words that contain three and more than 3 syllables in three 10-sentence samples from the text. It is suggested to be used on text that contains more than 30 sentences. The formula is shown below.

$$The\ SMOG\ Index = 3 + \sqrt{polysyllabic\ count}$$

In the formula, the polysyllabic count is defined as below:

1. Sample ten consecutive sentences near the beginning of the text, 10 in the middle, and ten near the end, totaling 30 sentences
2. Counting every word with three or more syllables and add them together

The SMOG Index should be interpreted with respect to Table 2.3.

The Gunning Fog Index[43] is calculated based on average sentence length and the percentage of complex words. The formula is shown below. Complex words here are defined as those containing three or more syllables. Its values normally fall in the range between 0 and 20. Values indicate the education level the reader

Value	School level	Student age range	Notes
0-1	Pre-kindergarten - 1st grade	3-7	for those who just learn to read books.
1-5	1st grade - 5th grade	7-11	Very easy to read.
5-8	5th grade - 8th grade	11-14	ideal for average readers.
8-11	8th grade - 11th grade	14-17	Fairly difficult to read.
11 and above	11th grade - college	17 and above	Too hard to read for the majority

Table 2.3: SMOG Index Interpretation

is expected to have to comprehend the text. If the text targets at the public, it should aim for an index around 8. Values should be interpreted with respect to Table 2.4.

$$\begin{aligned} Gunning\ Fog\ Index &= 0.4 * [ASL + 100 * PCW] \\ ASL &= Average\ Sentence\ Length \\ PCW &= Percentage\ of\ Complex\ Words \end{aligned}$$

Flesch Reading Ease[39] compute the readability score based on average sentence length and average word length. The formula is shown below. In the formula, the average word length is defined as the average number of syllables per word. Usually, the score ranges from 1 to 100, and 100 is the highest. If the text is scored between 70 and 80, it indicates the reader is required to have a school grade level of 8 and above. This also means the text is ideally readable for the average adult. Scores should be interpreted with respect to Table 2.5.

Value	School level	Student age range	Notes
0-1	Pre-kindergarten - 1st grade	3-7	for those who just learn to read books.
1-5	1st grade - 5th grade	7-11	Very easy to read.
5-8	5th grade - 8th grade	11-14	ideal for average readers.
8-11	8th grade - 11th grade	14-17	Fairly difficult to read.
11-20	11th grade - college	17 and above	Too hard to read for the majority of readers.

Table 2.4: The Gunning Fog Index Intepretation

$$\begin{aligned} Readability\ Ease &= 206.835 – (1.015 * ASL) – (84.6 * ASW) \\ ASL &= Average\ Sentence\ Length \\ ASW &= Average\ number\ of\ syllables\ per\ word \end{aligned}$$

The New Dale-Chall formula[55] use a predefined set of "common" words[6] that are considered familiar to fourth-graders to measure the readability of text. The more unfamiliar words contained in the text, the higher education level the reader is required to comprehend the text. The formula is shown below. This metric is only available for English due to the predefined English word set. Values refer to the US grade level and should

Value	School level	Student age range	Notes
100.00 - 90.00	5th grade	11	Very easy to read.
90.0 - 80.0	6th grade	11-12	Easy to read.
80.0 - 70.0	7th grade	12-13	Fairly easy to read.
70.0 - 60.0	8th grade - 9th grade	13-15	Standard, plain English.
60.0 - 50.0	10th grade - 12th grade	15-18	Fairly difficult to read.
50.0 - 30.0	College	18-19	Difficult to read.
30.0 - 0.0	College graduate	22-23	Very difficult to read.

Table 2.5: Flesch Reading Ease Interpretation

[6] Dale-Chall common word list: https://www.readabilityformulas.com/articles/dale-chall-readability-word-list.php

be interpreted according to Table 2.6. Research like [98] suggests that The New Dale-Chall is the most suitable metric for evaluating and selecting text material for the students.

$$\begin{aligned} New\ Dale\ Chall\ Score &= 0.1579 * (PDW) + 0.0496 * ASL \\ PDW &= Percentage\ of\ Difficult\ Words \\ ASL &= Average\ Sentence\ Length\ in\ words \end{aligned}$$

Value	School level	Student age range	Notes
4.9 or lower	Pre-kindergarten - 4th grade	3-10	Very easy to read.
5.0 - 5.9	4th grade - 6th grade	10-12	Easy to read. Conversational English.
6.0 - 6.9	6th grade - 9th grade	12-14	Fairly easy to read.
7.0 - 7.9	9th grade - 10th grade	14-16	Standard, plain English.
8.0 - 8.9	10th grade - 12th grade	16-18	Fairly difficult to read.
9.0 - 9.9	12th grade - college graduate	18-22	Difficult to read.
10 and above	University graduates	22 and above	Very difficult to read.

Table 2.6: New Dale-Chall Interpreation

2.5.1. Conclusion

There is very rare literature on the intersection of readability and people with visual challenges. However, this research [125] implicates that the use of short words and the inclusion of more familiar words can improve readability and comprehension for people with dyslexia. This finding is inspiring. The SMOG Index, The Gunning Fog Index, and Flesh Reading Ease involve syllables counting in computation, though in different ways. That means these three metrics favor text with short words. When it comes to including more familiar words, The New Dale Chall can be helpful since it uses a set of predefined common words to evaluate. In addition, the New Dale Chall is reported to be more suitable for students[98], which exactly coincides with our target audience. Therefore, we select these four metrics to evaluate the readability of sentences from our models.

2.6. Technology for the visually impaired and blind people

There are many people in the world suffering from visual challenge. The statistics from the World Health Organisation (WHO) shows globally there is more than 2.2 billion people who have a vision impairment. In the Netherlands, the estimated population who suffer from visual challenge reaches 32 million by 2020, of whom, up to 45 thousand people are blind. With the population growth and ageing, the figure of people acquire vision impairment may increase.

Visually impaired and blind people encounter many challenge in daily life. Not as the majority of the non-blind community reckon, the blind people or the visual impaired can lead a normal life in their own way of doing things. Indeed, they meet difficulties in daily life because of different barriers like inaccessible public infrastructure, unreachable information and etc.. Investigating into those barriers can help us understand them and using new technology might be helpful to eliminate barriers that hinder them gain independence.

Researchers in the field of assistive technology (AT) are devoted to help individuals with different impairment and the elderly people who need increase the quality of life. In the following subsections, we firstly give a broad view of assistive technology (AT) for people with vision challenge. Then we introduce assistive solutions that are especially based on smartphone.

2.6.1. Assistive technologies

Assistive technology for the visually impaired and blind people is aimed at helping them live independent like other people in the society by reducing physical, social, infrastructural and accessibility barriers. “technologies, equipment, devices, apparatus, services, systems, processes and environmental modifications” are all or partially involved in the field of assistive technology for the visually impaired and blind people.[49] It is such a complex flied with big scope varying from psychological phenomena, medical intervention to computer science technique. It can not be simply covered by few paragraphs.

There are many excellent survey on assistive technology for the visually impaired and blind people with different scope and emphasis: the complementary review across multiple disciplines by Bhowmick and Hazarika; the surveys on smart-phone based solution by Khan and Khusro and Hartato et al.; the research on mobile and wearable devices[10]; etc.. As far as we know, there is no review from the perspective of computer science technology on assistive technology for people with vision challenge yet. Here, we provides a summary mainly focus on assistive technology in the area of computer science as below.

- Computer vision. Vision substitution solutions collect data from a user’s environment and extract information from a variety of sources, including images, captions, tags, visual codes, and stickers. This extracted information is conveyed to the user through auditory or haptic feedback, enhancing their ability to perceive the surroundings. Numerous vision substitution systems and techniques have been developed for specific applications, such as navigation and shopping. Recently, there has been an increased focus on advanced functionalities, particularly in facial recognition, object recognition, and access to printed materials. These developments aim to augment the capabilities of computer vision, ultimately improving accessibility and independence for individuals with visual impairments
- Semantic augmentation. Blind individuals encounter considerable difficulties in perceiving unfamiliar spaces, landmarks, and points of interest. Effective navigation requires robust spatial modeling of the surrounding environment. A variety of representations have been developed to aid in spatial modeling, including geometric models, graph theory, symbolic representations, and hybrid approaches. Geometric models employ a coordinate system to facilitate queries such as locating roundabouts, while symbolic models utilize symbols to convey relationships and semantic meanings among different elements. The purpose of semantic annotation is to enhance the understanding of spatial semantics and navigational context, allowing for more informed responses to changes in the environment. Blind individuals face substantial challenges in navigating landmarks and avoiding obstacles. Enhancing the accessibility of navigational information for blind users is therefore crucial for improving their spatial awareness and overall independence.
- Augumented reality. Augmented reality has made significant strides in the past decade, primarily driven by improved processing capabilities and the incorporation of various sensors and multimedia features. Modern smartphones, equipped with high-resolution cameras, touch screens, and sensors such as accelerometers, GPS, and compasses, serve as comprehensive tools for blind users. These devices are powerful enough to support audio and echolocation functionalities enhanced by allied sensors. Although most augmented reality applications focus on visual modalities—overlaying virtual objects onto real-world scenes via live camera feeds—blind users benefit predominantly from auditory or hap-

tic feedback. Audio augmented reality is particularly useful for helping blind individuals perceive their environment through audio interfaces.

- Text to speech. Enhancing accessibility is a critical need for blind individuals. Screen reading applications offer significant assistance by converting text to speech, although they do have some limitations.

2.6.2. Smartphone-based assistive solutions

With the advance of mobile computing and sensors technology, smartphone is not just a simple communication device but also a carrier of diverse services. As a large number of visually impaired and blind people are using smart phone, researchers have investigated the opportunity in smartphone to provide blind-friendly technological solutions. In the following sections, some smartphone-based assistive solutions in different fields are illustrated.

- Navigation.BlindSquare functions as a GPS navigation tool that employs voice feedback to inform users about their current location and the surrounding area. It provides directions to specified destinations, announces nearby points of interest, and integrates smoothly with other apps, such as Uber and Google Maps. Furthermore, the application features extensive customization options to meet individual preferences and requirements.
- Accessibility. Similar to BlindSquare, Lazarillo places greater emphasis on public transportation and accessibility. It provides information on the nearest bus stops, subway stations, and bike-share locations, along with real-time updates on arrival times and routes. Additionally, it offers details about nearby accessible venues, such as restaurants, hotels, and ATMs, allowing users to rate these locations based on their accessibility features.
- Computer vision. Seeing AI is a free application that serves as a multifunctional tool for individuals with visual impairments, integrating various features into a single, user-friendly interface. Its primary function is to provide detailed descriptions of the surrounding environment, enabling users to recognize text, objects, people, and colors. Additionally, the app can read text aloud, describe scenes in photos, and identify currency. Although there may be a slight learning curve, the app is generally intuitive and proves to be exceptionally useful once users become familiar with its functionalities.
- Semantic augmentation. OKO is an application that utilizes artificial intelligence to detect pedestrian signals, facilitating easier navigation for blind and visually impaired individuals at intersections. Beyond simply indicating the current status of signals, the app also assists with orientation, making it an invaluable resource for users with visual impairments. It functions directly on the mobile device, ensuring immediate response times and allowing use in airplane mode if desired. The reliability of OKO has contributed to its rapid growth and increasing popularity among users.
- Augumented reality. Aira is a mobile application that offers users on-demand access to a team of certified agents who assist with everyday tasks through live video streaming and augmented reality. With the app, users can connect to Aira agents, who provide visual information and guidance for various activities, such as navigating unfamiliar environments, reading labels, identifying objects, and even online shopping. The agents leverage live video streaming and augmented reality technology to visually access the user's perspective and offer real-time assistance.
- Others. Be My Eyes is a mobile application that connects blind and visually impaired individuals with sighted volunteers for real-time visual assistance, enhancing user independence in daily activities. The app utilizes the smartphone's camera to stream live video to a volunteer, who aids with tasks such as reading labels, identifying objects, and providing directions. Available in over 180 countries and translated into more than 180 languages, it serves as a global platform for accessibility. The application has received significant recognition for its innovative approach to promoting inclusivity, winning awards such as the 2018 Google Play Award for Best Accessibility Experience and the 2017 AppleVis Golden Apple Award for Best Assistive Technology. Its user-friendly interface facilitates seamless interaction between visually impaired users and volunteers. Upon requesting assistance, users are matched with a volunteer who speaks their language.

2.6.3. Discussion

Based on the reviewed research, no existing tool or technology can be deemed ideal for assisting individuals with visual impairments. Consequently, it is essential to develop more sophisticated systems that directly address ongoing challenges. A key aspect of this development should be the involvement of intended users in the design, development, and evaluation processes of these systems.

- Integrating user interface modalities into the design and development process is essential for enhancing system accessibility and usability. Smartphone-based assistive aids can significantly improve in usability, accessibility, learnability, and adaptability by implementing effective user interfaces and advanced human-computer interaction techniques. Since interactions and body awareness are fundamentally multimodal experiences, assistive technology interfaces for visually impaired persons (VIP) should adopt this multimodal approach, allowing customization according to user preferences. By facilitating intuitive interactions through a combination of visual, auditory, and haptic feedback, the design can empower VIPs and improve their overall engagement with the technology.
- It is essential to distinguish between the needs of visually impaired persons (VIPs) who retain some vision or experience color blindness and those who are completely blind. This differentiation may create new research opportunities to address challenges beyond total blindness. Furthermore, the difficulties encountered by individuals with partial vision impairments and other chronic conditions, such as diabetes, have been insufficiently addressed in the existing literature.
- We propose that future assistive tools and technologies should capitalize on technological advancements to develop globally accessible navigation aids. These innovations should prioritize the creation of new concepts that employ low-cost technologies, efficient algorithms, and low power consumption. By focusing on affordability and sustainability, these navigation aids can become more widely available and effective, enhancing user experience and independence for individuals with visual impairments.

2.7. Summary

Although video captioning has achieved evident progress so far, there is still a long way to get optimal output. In the following paragraphs, we include challenges that exist in the video captioning task and point out the possible trends in the future based on the research.

Domain gap
The major research work in the video captioning task employs action recognition datasets like Kinetics-600[57] for pre-training and then uses the video captioning dataset for fine-tuning. The difference between datasets (in terms of quality, diversity quantity, etc.) for pre-training and fine-tuning will result in the domain gap.

Modality gap
Video captioning transforms video into text, which means it is a process to map continuous data to discrete data or a process to convert multi-modal data (image, audio) to single-modal data. In other words, the visual data and textual data are supposed to be mapped to the shared space. Obviously, there is a gap between the input and output. There is some research on leveraging the correspondence between textual and visual data.[112] [84] The latest research[80] firstly describes the modality gap generally and provides some qualitative and quantitative analysis on how the modality gap affects downstream tasks. However, they point out that "it's not clear that it's desirable to have no modality gap" [80]. In some tasks like zero-shot learning, having a larger modality gap may help fairness. Whether Modality gap have positive or negative impact on video caption is unknown, which need future study.

Insufficient dataset
It seems there are plenty of datasets for video captioning tasks. However, the truth is datasets available are far from sufficient with respect to several dimensions:

- Diversity. According to the research, the majority of caption datasets is fine-grained, centring on certain domain like movie[97], cooking[131], surveillance [31] and etc.. There are two open-domain datasets, MSVD [13] and MSR-VTT dataset [171], which are widely used. However, they only have a small number of classes. For instance, MSR-VTT only has 20 categories. A dataset with a large number of categories will help the video captioning model work better. The deeper the model is, the more diverse data is needed for training and avoiding over-fitting.
- Quality. The quality of the video and captioning are crucial for video captioning tasks. As a cross-modal task, the video captioning task brings CV and NLP together. Image quality is always one of the most important problems in CV tasks. Most captions in the existing datasets are collected by crowdsourcing, the quality of which is difficult to ensure homogeneousness. Better quality of the dataset will result in less noise and bias.
- Quantity. With the development of deep learning, the architecture of video captioning models become deeper and deeper. To some extent, the model requires vast data to be trained with to facilitate its potential capacity. The number of videos and captions in the video captioning dataset may not ensure models converge gradually.

Lack of explicit metric
So far, most accuracy measure metrics for video captioning models are originally designed for machine translation tasks and image captioning tasks. For instance, BLEU[109], ROUGE_L[82], METEOR[8] are from machine translation task; SPICE[2], CIDEr[155] are proposed for image captioning task. Since there is no explicit metric designed for video captioning task, the evaluation of model is considerably limited. The mainstream way is to score the model across different metrics and then analyze some samples that somehow depend on human evaluation. There is some effort to utility adversarial learning to optimize a metric[176]. Very recently, just after the emergence of Bert[30], the research community started to study the possibility of applying Bert for better evaluation for text generation [181] and image caption[71]. Lee et al. combine Bert and adversarial learning together and put forward an unreferenced metric for image captioning. However, related research is rare and immature. The lack of standard measuring metric, to a certain degree, has hindered the development of the video captioning model.

Efficiency
The efficiency problem is comparatively the most critical challenge that video captioning holds for several

reasons. The process of translating video to textual data is not as easy as putting a video into the model and then the text is ready. The video has to go through several preprocessing steps(like sampling, feature preparation, etc.) followed by the video representation formation and sentence generation. That's why there is merely no robust video captioning model that process realtime video and generate captions online.[53] Secondly, with the advance of research and the pursuit of performance, the model structure becomes deeper, and accordingly, the parameters scale grows bigger. This calls for huge computing powers. For instance, the transformer-related captioning model is outstanding for its performance and time/space complexity. Considering the current computing power and the video preprocessing procedure, the SOTA video captioning model is hardly possible to apply in practice.

3

User study

This chapter aims to answer the Research question: What are the requirements for video captioning for the visually impaired? In this part, we give details of how we prepare and conduct the user study.

3.1. Survey design

In order to answer this research question, we conducted a survey with the people in Visio twice. The first questionnaire is designed to get basic information about the visually impaired people at Visio and their dilemmas in life. The second questionnaire is designed based on the information collected from the first one. With the purpose of further verifying the needs of the target audience, we propose a scenario: There is an app on the mobile phone. They can use it to get some information about the environment they are in or things that happen nearby by taking a video of their surroundings (the algorithm behind the app is what we aim at). From this survey, we got some requirements that helped us design the video captioning model afterward.

3.2. Survey result

In this subsection, the demography of the people in the Visio is presented. The difference between interviewee is addressed as well. Besides, we form a list of requirements that we concludes from the surveys, which decides the research direction of this thesis. Some interesting points are further discussed in the end of this chapter.

3.2.1. Demography

Given that Visio at Grave is an institution that provides education and support to students with visual impairment, the audience of our survey is teenagers in the age range of 16 to 19 years old. In total, there were 5 students who participated in our survey. There is a difference between students who are blind their whole life and students who became blind at an older age. Most of these blind students see a little bit (useful for mobility and shapes). As most of them are not good at English, their coach serves as the interpreter and coordinator for the whole procedure. For comparison, we also involved senior people over 60 years old in this survey. The overall demography is shown below:

Table 3.1: Demography

Name	Age	Gender	Sight	saw before
Person 1	16	M	blind (low sight. Has the best vision of this group)	
Person 2	17	M	blind (low sight. became blind on an older age. saw pretty good)	✓
Person 3	19	F	blind (very low sight. Useful sight for mobility, shapes)	
Person 4	18	F	blind (very low sight. Useful sight for mobility, shapes)	
Person 5	16	M	blind (very very low sight. Useful sight for mobility, shapes)	
the Senior	>60	M	blind(low sight. became blind on a older age. saw pretty good)	✓

From the table 3.1, we can see that 6 participants are well selected. Students of different sight levels are chosen, ranging from low, very low to very very low. (The coach said they belong in the blind category. Thus, we call them blind in the table.) As we observed, people who can see before and people who are born to be blind have significantly different preferences when filling in the questionnaires. We involved a student who lost sight at an older age in the survey. To see how blind people of different ages differ, we also invited senior people to answer the questionnaires.

3.2.2. Insight

Based on the data we collected from two questionnaires, we formed a list of requirements as below. As we gave interviewees a scenario: There is an app on the mobile phone. They can use it to get some information about the environment they are in or things that happen nearby by taking a video of their surroundings. Questions were asked on the foundation of this scenario.

Table 3.2: List of Requirements

Num	Question	Answer
1	How long will the video take?	no longer than 30s
2	How many seconds of latency can you accept at most?	limit to 3-4s
3	When will you use this system for help?	when there is loud music/noise
4	What do you want this system tell you?	a wide variety of topics but care actions most
5	How do you prefer the system to describe the video?	by a single sentence briefly

Regarding some questions, interviewees had different opinions. All the requirements are formed based on their consensus. Each requirement is further explained in the following section.

Requirement 1: The video should not be longer than 30s. Interviewees gave different answers to these questions, from a few seconds to 1 minute at most. Therefore, to form the first requirement, we take the average 30s seconds.

Requirement 2: :The latency should be limit to 3-4s. The latency here is defined as the time interval between the user finishes taking the picture/video, and the system starts to describe it. Answers to the latency didn't vary much. Interviewees said they could wait longer if more information is needed.

Requirement 3: Scene: when there is loud music/noise. From the survey, we learned that visually impaired and blind people could use accurate hearing to recognize things that happen around them, not as the majority of people reckon. For example, they get orientation from hearing echoes from walls. If there is a lot of noise, they can't recognize things. Loud bikes, cars, trucks, and buses are most of the time the problem. Thus, they tend to use this system when there is loud music/noise.

Requirement 4: The dataset should cover a wide variety of topics, but the actions that happen are most important for them. When we asked the interviewees what they expected the system to tell them, they came up with many answers. They want to know the details about their surroundings, information about people (where they are, how many are there, how someone looks), the appearance of something, and so on. They also wish the system could help them cook. One of them said he does not care about people unknown. All of them agreed that they care more about What people are doing (human action). Based on this finding, we realized that they expect a system that works for the open domains. To verify this assumption, we asked interviewees to pick up their topics from people, sports/actions, vehicles/autos, howto, travel, animals/pets, kids/family, food/drink, cooking, beauty/fashion, and advertisement. Most of the topics were picked. There was a small controversy on topics such as people, kids, beauty, and science. To cover the interviewees' needs, we have included all the topics in this thesis work.

Requirement 5: The video should be described in a single sentence briefly. The purpose of this question is to know what kind of output the visually impaired and blind people expect from the system. Because the type of descriptive output determines different research directions in the field of video captioning. Most interviewees want the system to describe the video in a single sentence briefly and in a very detailed, story-telling way. However, when it comes to latency (the more detailed description requires more time to generate), interviewees tend to use brief descriptions. In most cases, they expect responsive descriptions.

Interesting findings

There are a few things that are worth mentioning here:

1. There is a difference between interviewees who are blind since they born and interviewees who became blind on an older age. Interviewees who are born-blind show less interest in the realistic world. This

is probably because they do not establish awareness of many objects and events in their mind without useful sight at an early age. In contrast, interviewees who became blind at an older age are more curious about events in their surroundings and are eager to know more.

2. Counter-intuitively, most of the interviewees have laptops, phones, iPads, and even PlayStations. They prefer the iPhone to Android as the iPhone is more blind-friendly. They use social media(like Instagram and Facebook), agenda, digital paying, email, and so on.
3. From the survey, the interviewees show many needs in real life, such as recognizing friends, reading text, recognizing surroundings, etc. These needs lead to different research fields or tasks like facial recognition, text recognition, object recognition, and so on. Their needs are complex and cannot be met solely with algorithms or models as far as current technology goes.

In summary, we interpret the five requirements briefly here: The interviewees expect a system or algorithm that can describe the video in less than 30s by a single and brief sentence with latency limited to 3-4s (the latency here is defined as the time interval between the user finish taking the picture/video and the system starts to describe it). The algorithm should work on many topics, which indicates that the dataset used for training should be open-domain. Human actions are the most important topic that needs to be paid attention to. The interviewees tend to use this system when there is a lot of noise. This means the algorithm should be robust to noise, or the audio should not be suitable as input. The noise may lead to bias in real life.

4

Model Design

In this chapter, we propose a video captioning model for the visually impaired based on the mainstreaming encoder-decoder framework but with modified temporal attention and designed features. As shown in Fig. 4.1, we present an overview of the model, which can be briefly decomposed into three parts from left to right: Frame sampling, feature extraction (Encoder), and language model (Decoder). Firstly, the video is split into n equal-sized segments, and then a single RGB frame is sampled from each segment. The selected frames are fed into three different pretrained dnns (Resnet, ECO, and Res3D) for features of various levels, ultimately yielding a representation of the video by concatenation. Ultimately, the video representation is decoded by the RNN-based language model with an attention mechanism in text. From the user study, we know that the visually impaired more care about the actions in the video content and the latency of the model. Thus, we tailor our design to their needs by emphasizing capturing video dynamics and reducing time costs. Besides, we decide not to use the audio information. In the following sections, the details of the model are introduced.

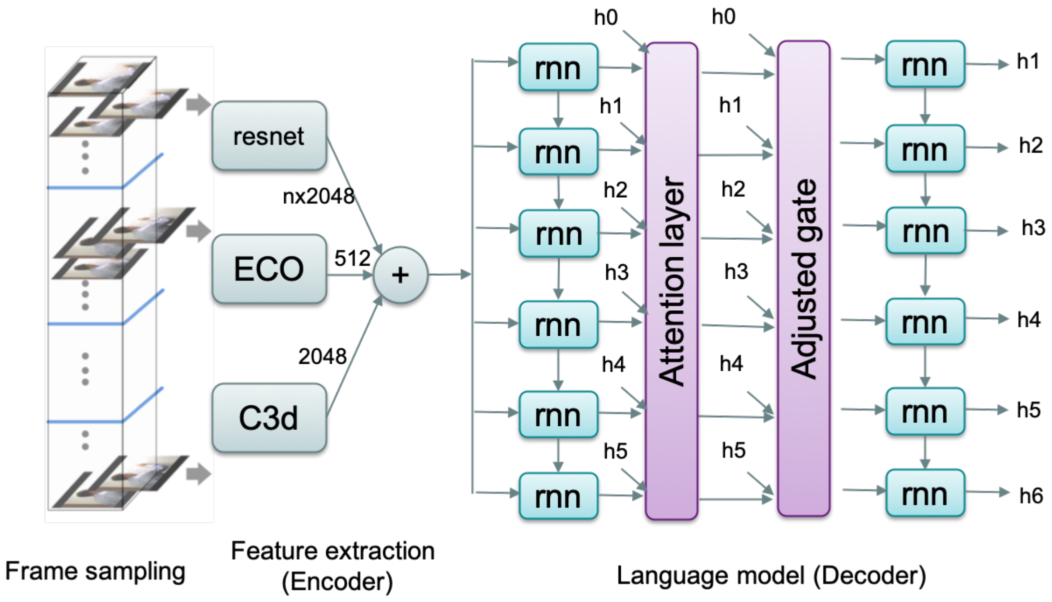


Figure 4.1: the video captioning model for the visual impaired is designed in this thesis.

4.1. Sampling strategy

In our model, we adopt the sparse temporal sampling strategy proposed by Temporal Segment Network(TSN) [161] and its later work Efficient Convolutional Network (ECO)[188] in the field of action recognition. As

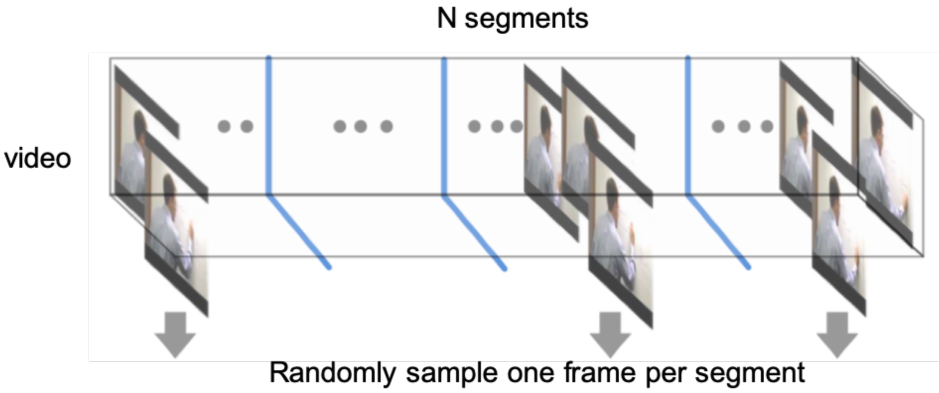


Figure 4.2: Sampling strategy. figure is adopted from [188].

shown in Fig. 4.2, the entire video is split into n equal-sized segments, and then a single frame is randomly sampled from each segment. As a result, a stack of selected frames awaits to be processed in the next step. The underlying concept is that the information in adjacent frames in the video stream is redundant, thus randomly sampling fixed number of frames from evenly split video segments not only ensure the coverage of long-range temporal structure of video but also reduce the computation cost in later processing. Unlike TSN and ECO, which employ a so-called two-stream architecture (RGB stream and optical flow stream), we only use RGB stream here to avoid the heavy time and computation cost of extracting the optical flow from the video. In this thesis, we sample 16 frames per video.

4.2. Encoder-Decoder framework

The encoder-decoder framework is widely applied in the open-domain video captioning task, as we already introduced in the section 2.2.2.1. Most approaches use pretrained CNN on ImageNet to extract frame-level features and optical flow or C3D to capture video dynamics during the encoding stage. Then, these features are simply concatenated or fused as the video representation. The learned video representation is later passed through the decoder (usually using RNNs like GRU or LSTM) to generate captions. Our proposed model follows this framework as well. Still, we are more focused on retaining the motion in the video content, from capturing local motion features during the encoding stage to obtaining the global temporal structure of the video during the decoding stage.

4.2.1. Feature extraction (Encoder)

During the encoding stage, we use three pretrained deep neural networks (Resnet, Res3D, and ECO) to extract frame-level feature, long-term action feature, and short-term dynamics, respectively, and then concatenate them to form the final video representation.

Frame-level feature

A residual network, namely Resnet[47], was designed to deal with the degradation problem of the deep neural networks caused by the gradient vanishing via introducing an identity shortcut connection. In the left of Fig. **??**, a Resnet block is presented: Given x as the input, $H(x)$ is the underlying mapping for the stacked layers. While [47] applies a shortcut connection by a simple identity mapping presented as a bypass in the figure and adding its outputs x to the outputs of stacked layers. Intrinsically, now this block approximates a residual mapping $F(x) := H(x) - x$, instead of $H(x)$. Thus, the original function is converted to $F(x) + x$. That's where the main difference between ResNet and traditional DNNs lies. Traditional DNNs learn a mapping of $H(x)$ directly; ResNet forces itself to understand the residual of input x and output of the stacked layers. This enables the gradient to flow back when the number of layers increases. That is, when $F(x) = 0$, Resnet can still pass x back as $H(x) = x$. In other words, Resnet forwards the gradient from higher layers to lower layers without any modification.

Once Resnet was proposed, it set new benchmarks for image recognition tas,k, and more and more variants emerged. Besides, lots of pretrained Resnet and its variants on ImageNet are available. We use the pretrained Renets provided by **torchvision** [1] to extract visual features. Given a sequence of n selected frames, $F = f_1, f_2, ..., f_n$, each frame image is scaled to 224×224 pixel and then is fed into the pretrained Resnet to get 2048-dimensional feature vector x. As a result, we get a stack of visual features $X = x_1, x_2, ..., x_n$. As observed in [180], Resnet shows its potential to scale up to thousands of layers with increased performance. However, it is at the cost of adding up the numbers of parameters largely to train. In the table 4.1, we summarise the most popular Resnet with a different number of layers 4.1 and their performance on the ImageNet. From the table, we can see that one percentage decrease in error rate approximately requires doubling the number of parameters. In order to make a trade-off between performance and efficiency, we compare them in the experiments.

Table 4.1: Error rates (percentage) of Resnet with different layers on the ImageNet validation set.[47]

Model	# params	top-1 err.	top-5 err.
resnet34	21.8M	21.53	5.60
resnet50	25.6M	20.74	5.25
resnet101	44.5M	19.87	4.60
resnet152	60.2M	19.38	4.49

Short-term dynamics

C3D[151] stands for 3D convolutional network, a simple yet effective approach for learning spatiotemporal features in the short term. Different from 2D ConvNets that apply 2D convolution on videos and result in a 2D feature map, C3D generates feature volume for a video via small $3 \times 3 \times 3$ convolution kernels, preserving temporal information. To explore how effective deep CNNs with 3d kernels work on the existing video datasets, [45] make experiments on a bunch of 3D CNNs covering comparatively shallow networks to very deep ones. For example, they propose to integrate Resnet (or its variants) with the 3D architecture, namely Res3D. There are several interesting findings: 1) the Kinetics dataset is capable of training deep 3d networks such as 3d ResNets of 152 layers without over-fitting. 2) The performance of 3d ResNets on the Kinetics dataset is similar to the 2D ResNets on ImageNet. 3) The ResNeXt-101[170] achieved the best accuracy, and its pretrained model outperforms other networks on UCF-101[138] and HMDB-51[64] datasets.

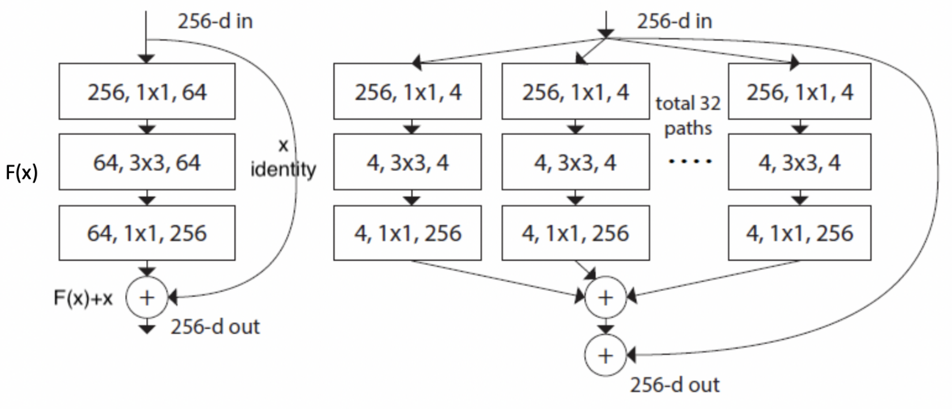


Figure 4.3: The architecture of ResNeXt compared with the architecture of Resnet.
Left: a Resnet block. Given x as the input, $H(x)$ is the underlying mapping for the stacked layers. While [47] applies a shortcut connection by a simple identity mapping presented as a bypass in the figure and adding its outputs x to the outputs of stacked layers. Intrinsically, now this block approximates a residual mapping $F(x) := H(x) - x$, instead of $H(x)$. Thus, the original function is converted to $F(x) + x$.
Right: a ResNeXt block. A block of ResNeXt has several branches, which are formally called transformations. All of them share the same topology, and their outputs are aggregated by summation. In other words, ResNeXt extends Resnet with a new dimension that is defined as "cardinality" (the number of transformations). Each transformation is on low-level embedding, whose outputs are aggregated together by summation. Similar to Resnet, a shortcut connection is employed as well. This figure is adopted from [170] and modified.

Following these interesting findings, in this thesis, we choose 3D ResNeXt101 to extract short-term dynamics from videos. ResNeXt-101[170] is a variant of Resnets of 101 layers that adopts the aggregated transformations inspired by the split-transform-merge concept of Inception (GoogLeNet)[141]. As shown in the

[1] https://github.com/pytorch/vision

right of Fig. 4.3, we can see that a block of ResNeXt has several branches, formally which is called transformations. All of them share the same topology, and their outputs are aggregated by summation. In other words, ResNeXt extends Resnet with a new dimension that is defined as "cardinality" (the number of transformations). Each transformation is on low-level embedding, whose outputs are aggregated together by summation. Similar to Resnet, a shortcut connection is employed as well. As reported in [170], although ResNeXt has more complex architecture than Resnet, owing to the homogeneous branches, the number of hyper-parameters needed to be set is approximately the same as Resnet, and it can outperform Resnet in some tasks.

We adopted the pretrained 16-frame ResNeXt model that achieves the best accuracy on the Kinetic dataset in the video classification task from [45][2]. It has 101 layers and 32 cardinality. For the short connection, it uses shortcut Type B [47] that adds an extra convolutional layer in the identity shortcut path to solve the dimension mismatch between the input and output. Then, we finetune its Conv5x and fc layers on our training set until it converges (about 25 epochs) and uses the finetuned model to extract short-term dynamics from the videos. More specifically, for each input video, we implemented the spatial transform and the temporal transform. Temporal transform aims to split the video into non-overlapped n 64-frame segments (if the segment is less than 64 frames, we loop itself as many times as necessary). Spatial transform is to crop every frame image in the segments around a center position to 112x122 pixels. Accordingly, the size of each sample is 3x64x112x112 (channel x frame x pixel x pixel), and the output from the pretrained model for each video is a stack of 2048-dimension features. In the end, we take the arithmetic mean along the axis of the stack, which results in a single 2048-dimension context vector.

Long-term action features

Although Res3D shows its capacity in the action recognition task, it cannot cover the temporal information of the entire video. To deal with this problem, we introduce ECO(Efficient Convolutional Network for Online Video Understanding)[188], to abstract the most significant action overtime and capture the relationship among frames. Fig. 4.4 presents the overall architecture of ECO. Firstly, we can see that ECO uses the same aforementioned sampling strategy in section 4.1, dividing the whole video into N equal-sized segments and randomly selecting one single frame from each of them. The sampled frames are fed into a 2D convolutional network separately, the weight of which is shared. Accordingly, each frame is encoded into a Kx28x28 feature map for the scene. Later, all of them are stacked together(KxNx28x28) and passed through a 3D convolutional network to learn the temporal relationship of frames and trace the action in the scene over time. ECO has a compact and straightforward architecture, which naturally can be easily adapted from its original field (Action recognition) to other video understanding tasks like video captioning.

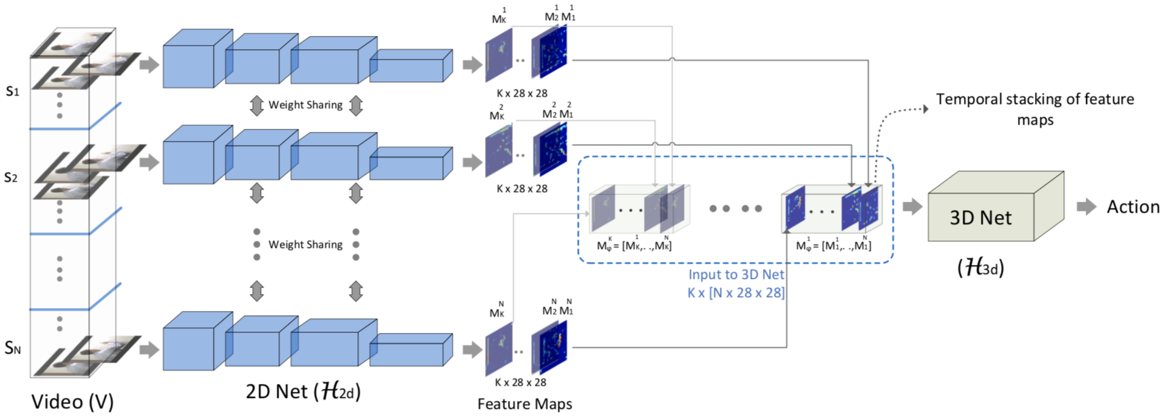


Figure 4.4: firstly, we can see that ECO uses the same aforementioned sampling strategy in section 4.1, dividing the whole video into N equal-sized segments and randomly selecting one single frame from each of them. The sampled frames are fed into a 2D convolutional network separately, the weight of which is shared. Accordingly, each frame is encoded into a Kx28x28 feature map for the scene. Later, all of them are stacked together(KxNx28x28) and passed through a 3D convolutional network to learn the relationship of frames and trace the action in the scene over time. [188]

We adopted a 16-frame ECO Lite model that pretrained on the Kinetic dataset from [188][3]. Then, we

[2] For details, see: https://github.com/kenshohara/video-classification-3d-cnn-pytorch

[3] For details, see: https://github.com/mzolfaghari/ECO-pytorch

finetune the last fc layers of the 3D net on our training set until it converges (about 20 epochs) and use the finetuned model to extract long-term action features from the global pool layer of the model. More specifically, for each input video, we transform it 3x16x224x224 (channel x frame x pixel x pixel) and pass it through the pretrained model, which yields a single 512-dimension context vector as the long-term action feature.

Feature concatenation

In summary, we get three different kinds of features as presented in table 4.2. In order to achieve a final representation that encodes scene information with (short-term and long-term) dynamics for the video, we treat the features extracted from pretrained ResNeXt101 and ECO as the context and simply add them to the end-of-frame-level features, which leads to a representation of 16x4608 dimension. In the experiments, we further analyze how the context vector influences the model performance.

Table 4.2: Summary of features

pretrained model	Dataset	Fintuning	Feature dimension	Feature
Resnet	ImageNet	No	16x2048	Frame-level feature
ResNeXt101	Kinetics	25 epochs	2048	(short-term) local temporal feature
ECO	Kinetics	20 epochs	512	(long-term) local temporal feature

4.2.2. Language Model (Decoder)

When the video representation V from the encoder is obtained, we employ the language model to automatically decode it into a sentence $S = \{s_1, s_2, ..., s_T\}$ to describe the video content. Our language model (as shown in the right part of Fig. 4.1) adopts the sequence-to-sequence architecture together with two layers, temporal attention and sentinel gate. In the following sections, we illustrate how we modify them and integrate them together.

Sequence-to-sequence architecture

Before we explain the adopted sequence-to-sequence architecture, we briefly introduce its key component in the field of sentence generation, Recurrent Neural Network (RNN). RNN is frequently used due to its ability to recognize patterns in sequences of data. Different from feedforward networks, RNNs have a feedback loop connected to their past decisions, taking their output of the last time as the new input. Theoretically, RNN can preserve the sequential information and find the correlation between events that are separated at different times as long as possible. From another perspective, RNN is capable of sharing weights over time. However, in practice, training RNN relies on Backpropagation Through Time (BPTT) [166]. BPTT suffers from the vanishing gradient problem, which hinders RNN from learning the long-term dependency. Later, two variants of RNN were proposed to solve this problem.

- **Long Short Term Memory (LSTMs)**[50] is a variant of RNN, which is designed to avert the vanishing gradient problem so as to learn long-term dependencies. The key to LSTMs is a memory cell m_t, which is controlled by three gates (input i_t, forget f_t, and output o_t). The memory cell works like a carrier of information over time. i_t decides whether the input can change the state of m_t while f_t determines what to remember or forget by the cell. Gate o_t allows or prevents the cell state from impacting the other neurons. LSTMs are formulated as below:

$$i_t = \sigma(W_i y_t + U_i h_{t-1} + b_i) \tag{4.1}$$
$$f_t = \sigma(W_f y_t + U_f h_{t-1} + b_f) \tag{4.2}$$
$$o_t = \sigma(W_o y_t + U_o h_{t-1} + b_o) \tag{4.3}$$
$$g_t = \phi(W_g y_t + U_g h_{t-1} + b_g) \tag{4.4}$$
$$m_t = f_t \odot m_{t-1} + i_t \odot g_t \tag{4.5}$$
$$h_t = o_t \odot \phi(m_t) \tag{4.6}$$

y_t denotes the input to the LSTM at time t, and W, U, and b are parameters (the weight matrices for the input y_t, the previous hidden state h_{t-1}, and the bias, respectively) need to be learned. σ is the sigmoid activation function. In addition to i_t and g_t, the candidate information, as shown in Equation 5.4, is computed to determine what goes as the input into the memory cell when some more information is required. In Equation 5.6, ϕ represents the hyperbolic tangent function, and h_t denotes the new hidden state.

- **Gated Recurrent Unit(GRU)**[25] is a new generation of RNN, which is proposed to handle the short-term dependency and gradient vanishing problem as well. GRUs follow the same input/output architecture from RNN but adopt the underlying concept of LSTMs. GRUs remove the memory cell and use the hidden state h_t directly to transfer information. It only has two gates: a reset gate r_t and an update gate z_t. r_t works similar to i_t of an LSTM. It determines how to combine the new input with the previous memory (what input information to use or discard). z_t is used to decide how much of the previous memory h_{t-1} to keep around and can choose to forget and remember at the same time. GRU is formulated as below:

$$z_t = \sigma(W_z y_t + U_z h_{t-1} + b_z) \tag{4.7}$$

$$r_t = \sigma(W_r y_t + U_r h_{t-1} + b_r) \tag{4.8}$$

$$\hat{h_t} = \phi(W y_t + r_t \odot U h_{t-1} + b) \tag{4.9}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h_t} \tag{4.10}$$

  y_t denotes the input to the GRU at time t, and W, U, and b are parameters (the weight matrices for the input y_t, the previous hidden state h_{t-1}, and the bias, respectively) that need to be learned. σ is the sigmoid activation function. In Equation 5.9, $\hat{h_t}$ is introduced as a new memory content that uses r_t to preserve the relevant past information. h_t in Equation 5.10 denotes the new hidden state. It is also the final memory at the current time t, which is calculated to decide what to collect from current memory content $\hat{h_t}$ and what from the past h_{t-1}.
  From the equations, it is notable that GRU has fewer tensor operations than LSTM. Thus, it is comparatively easier to train and computationally cheaper than LSTMs. As reported in [25], GRUs have a similar performance to LSTMs in many applications. There is not a clear winner as to which one is better.

Back to the topic, we follow the Sequence-to-sequence architecture of Sequence to Sequence model - Video to Text (S2VT)[157] (which is aforementioned in Section 2.2.2.1). S2VT employs a two-layer stacked LSTM to generate descriptions of the video content. As presented in the left of Fig. 2.3, the first LSTM layer processes a sequence of visual frame features and encodes them into the hidden representations, while the second LSTM layer turns them out into the word sequence when it receives the beginning-of-sentence tag. S2VT naturally keeps the temporal structure of video, and it is able to generate sentences of variable lengths. Inspired by this work, we also use two layers of RNN for sentence generation, namely the top and bottom RNN. However, we combine it with the attention mechanism and compare how effective LSTM and GRU work in the experiments.

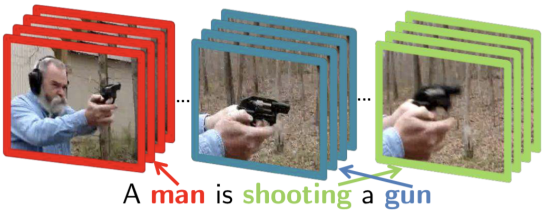


Figure 4.5: The illustration of temporal attention.[175]

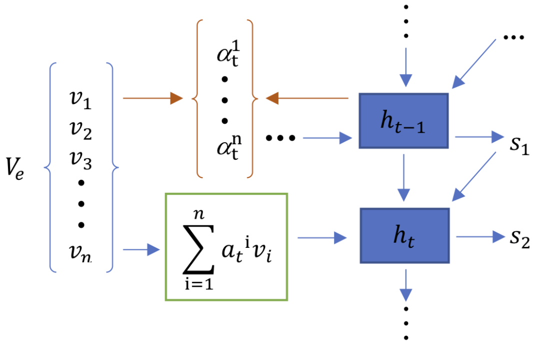


Figure 4.6: Temporal attention.[175]

Attention mechanism

We use res3D to obtain short-term dynamics and ECO to learn the relationship of actions that appear among frames, but both of them model the local temporal structure of the video. To capture the global temporal structure, we employ the Attention mechanism. Attention mechanisms in deep neural networks are inspired by human attention that sequentially focuses on the most relevant parts of the information over time. We adopt the temporal attention mechanism [175] to learn a mapping that guides the model to know which set of moments (frames) to look at when generating word sequences. As illustrated in Fig. 4.5, there is a video with its caption "A man is shooting a gun" where "man","shooting" and "gun" point to the collection of frames that contains the figure of themselves respectively. To generate a word like "man" the model ought to focus on the set of frames that contains the figure of man. Similarly, to generate "shooting" and "gun" the model needs to look at their relevant temporal regions.

Intuitively, temporal attention can assist this work. However, in the same sentence, words like "a" and "is" are less relevant to the content of the video, which means the model does not need to use the visual information from the video to generate such words. To handle this issue, on top of the temporal attention, we add an extra layer, namely the sentinel gate[93], that decides whether the model utilizes the visual information or relies on the context learned by the language model when generating words. In the paragraphs below, we explain how the temporal attention with the sentinel gate works.

- **Temporal attention** [175]
  Specifically, we use soft attention to capture the temporal structure of the video. Soft attention computes the gradient in the regular backpropagation way, which makes it easy to adapt to RNN. As shown in Fig. 4.6, the soft attention mechanism generates a vector of attention weights α_t^i for all selected frames at each time step t based on the previous hidden state h_{t-1} from the bottom RNN of our model (which presumably summarizes all the previously generated words) and the corresponding frame's temporal feature v_i from the top RNN. Instead of a simple averaging strategy used in [33], the dynamic weighted sum of the temporal feature vectors according to attention weights generated at each time step are fed into the bottom RNN (the actual caption generator). The attention mechanism makes the decoder capable of focusing on a certain subset of frames by increasing the attention weights α of the corresponding temporal feature V_e and vice versa.
- **Sentinel gate**[93]
  Lu et al. introduces the sentinel gate in the image captioning task to automatically guide the decoder when to look at the image and when to rely on the language model to generate the next word. They extend the LSTM structure with an additional (sentinel) gate (which is similar to other gates in LSTM), generating a sentinel vector other than a single hidden state.

$$\epsilon_t = w^T \phi(W_a \hat{h_{t-1}} + U_a V_e + b_a) \tag{4.11}$$

$$\alpha_t = softmax(\epsilon_t) \tag{4.12}$$

$$c_t = \frac{1}{N} \sum_{N}^{i=1} \alpha_t^i v_i, \quad where \quad \sum_{N}^{i=1} \alpha_t^i = 1 \tag{4.13}$$

Overall
In summary, With the aforementioned background information, our decoder can be formulated as below:

- **Top RNN layer**
  Input: the sequence of video features V output: encoded video features V_e and the hidden state h_t
- **Attention layer**
  using the hidden state of the top RNN at the last time step to initialize the hidden state of the bottom RNN
  Given the current hidden state of the bottom RNN layer $\hat{h_{t-1}}$ and the sequence of encoded video features V_e

$$\epsilon_t = w^T \phi(W_a \hat{h_{t-1}} + U_a V_e + b_a) \tag{4.14}$$

$$\alpha_t = softmax(\epsilon_t) \tag{4.15}$$

$$c_t = \frac{1}{N} \sum_{N}^{i=1} \alpha_t^i v_i, \quad where \quad \sum_{N}^{i=1} \alpha_t^i = 1 \tag{4.16}$$

- **Sentinel gate**
  Given the current hidden state of the top RNN layer $\hat{h_t}$

$$\hat{c_t} = \beta_t c_t \tag{4.17}$$

$$\beta_t = \varphi(W_s h_t + b_s) \tag{4.18}$$

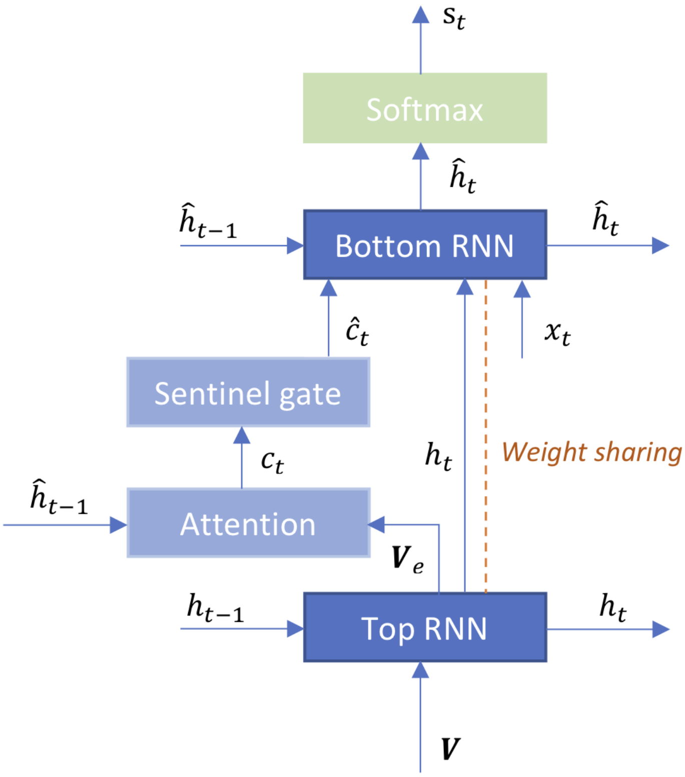


Figure 4.7: The illustration of language model.

β_t is projected into the range of denotes the parameters to be learned β_t is projected into the range of $[0,1]$. When $\beta_t = 1$, it indicates that full visual information is considered, while when $\beta_t = 0$, it indicates that no visual information is considered to generate the next word.

- **Bottom RNN layer**
  For the bottom RNN layer, one of the input x_t is the embedding vector of the previous word s_t at each time, given by:

$$x_t = W_e s_t \tag{4.19}$$

concatenate with the $\hat{c}_t$

$$\hat{x}_t = [x_t, \hat{c}_t] \tag{4.20}$$

- **MLP layer**
  The probability distribution over a set of possible words is obtained using the output of the Top RNN layer as

$$p_t = softmax(U_p \phi(W_p \hat{h}_t + b_p) + d) \tag{4.21}$$

4.2.3. Beam search

As we discussed in the last section, we get a probability distribution of words in the vocabulary through a linear MLP layer. To get the word s_t, the most straightforward and greedy approach would be to pick up the word with the highest likelihood and use it to predict the next word. Although this approach is simple and time-saving, it is not optimal. In essence, what we expect is the best sequence other than a set of words of the highest likelihood at each time. As the decoder generates the rest sequence relying on the previous word, we need to get the sequence of the highest overall conditional probability. Thus, we employ Beam Search[167].

Other than greedy search that decides word immediately, Beam Search considers several candidates each time and save a table of the candidate sequences. After the decoding finishes completely, the sequence that has the highest overall score from a basket of candidate sequences is selected. The number of candidate word is called beam width.

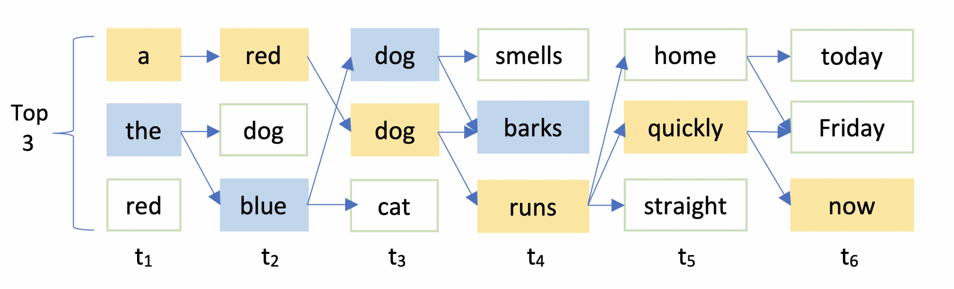


Figure 4.8: Beam search. figure is modified from [167].

The above figure gives an instance of how Beam search works with beam width 3. Vertically, the top 3 probable words are taken into consideration at each time t. Horizontally, each path presents a candidate sequence. Let us assume the blue chunks are the words picked by the greedy search: "The blue dog barks." This means that at each time t, only the word with the highest probability is selected. Searching over each path in the figure, "A red dog runs quickly now" in the golden chunk' is the sequence found by Beam search.

There are few thing to note. Theoretically, beam search strategy enlarges the search space, it helps to find a better solution. However, to search over all possible outcomes is time-consuming. Sometimes it may not result in a goal at all, therefore beam width is set to limit the search space and prevent the endless loop when a goal cannot reach.

The readers may observe the generated sequences are of different length. It is because during the sentence generation, a EOS token is used to suggest when the process can stop. Normally it would be 0. In our case, when a EOS token generated from the top RNN layer, the bottom RNN layer will stop decoding. Vice versa, a SOS token is used as well to tell the decoder to start decoding. In practice, the sentence length will be manually limited to prevent some extreme situations.

5
Experiment Setup

In this part, the dataset we use is going to be introduced first. We will give details of how we prepare the data and the procedure by which we train the models.

5.1. Data preparation

In this section, we explain with analysis how we select the dataset and what preprocessing procedure we take on the dataset before we train the model.

5.1.1. Dataset and Analysis

We use MSR Video-to-Text [171] to evaluate the effectiveness of our proposed model. It is an open-domain dataset that has the largest amount of clip-sentence pairs (200K in total). These videos are segmented into clips that last 10 to 30 seconds. Each clip is annotated with 20 independent sentences. All videos in this dataset are collected from the online search engine, covering 20 representative categories[1]. Altogether, it is 41.2 hours long and contains 29,316 unique words. B sides, this dataset encloses an audio channel, which may be used as the multimodal feature.

There are several reasons why we chose this dataset. First, there is no available video dataset annotated with descriptions for the visually impaired. Second, due to the pandemic and privacy issues, it is not possible to collect data within a short time. M R-VTT dataset is what we could find the most suitable one that meets the requirements of the visually impaired. From the User Study 3, we know that the target audience has several requests that influence the choice of dataset:

1. The dataset should cover a wide variety of topics. (people, sports/actions, vehicles/autos, howto, travel, animals/pets, kids/family, food/drink, cooking, beauty/fashion, advertisement)
2. The video should not be longer than 30s.
3. The video should be described in a single sentence briefly.

We analyse this dataset with respect to aforementioned three requirements. In the following paragraphs, we will explain why the MSR-VTT dataset meets the user requirements and is the best choice among several existing datasets. The MSR-VTT dataset is the best choice in our case compared with other datasets. Let's look into Table. 5.1.

In terms of diversity, M-VAD[148], MPII-MD[128], and TACoS-multi-level[126] are narrowed in the field of movie or cooking, Although MSVD[13] consists of videos which of different categories, its number of video (1970 in total) is far too less for training. In contrast, the MSR-VTT dataset contains 7180 videos of 20 categories that are crawled from a commercial video site. Fig. 5.1 presents the distribution of video categories in the MSR-VTT dataset. To ensure the representativeness of the dataset, MSR-VTT contains the top 150 videos for the top 257 representative queries regarding 20 categories[171]. us, this distribution reflect the real interest of people in daily life. The scale of MSR-VTT (7180 videos in total) ensures the diversity of video as well.

[1] These 20 categories include music, people, gaming, sports (actions), news (eventspolitics), education, TV shows, movie, animation, vehicles, how-to, travel, science (technology), animal, kids (family), documentary, food, cooking, beauty (fashion), advertisement.

Table 5.1: Comparison among different video-to-text datasets

Dataset	Context	Total videos	Total clips	Avg. clip length (s)	Total video (h)	Total sentences
M-VAD	Movie	92	48,986	6.2	84.6	55,904
MSVD	Various/open	1970	1970	10	5.3	70,028
MPII-MD	Movie	94	68,337	3.9	73.5	68,37
TACoS-multi-level	Cooking	185	14,105	360	27.1	5 52,593
MSR-VTT	Open	7180	10,000	20	41.2	200,000
ActyNet Cap.	Open	20000	20,000	180	849	100,000

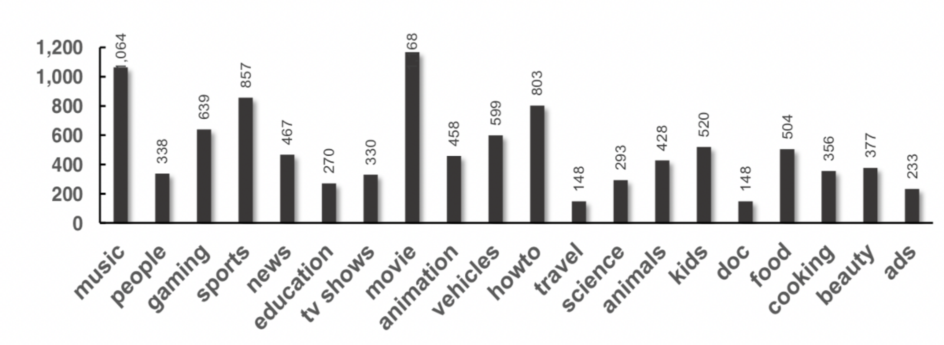


Figure 5.1: The distribution of video categories in the MSR-VTT dataset. Figure is adopted from [171]

Concerning the average clip length, across all datasets, only MSR-VTT (20 seconds) is close to 30 seconds, as required. The video length of other datasets is either too long or too short. The video lengths of M-VAD[148], MPII-MD[128], and MSVD [13] are all below ten on average. The figure of that for ActyNet Cap. [38], and TACoS-multi-level[126] are 180 seconds and 360 seconds, respectively, far from our 30-second requirement. To further analyze the MSR-VTT dataset, we make a box plot to see the distribution of clip length among different categories as shown in Fig. 5.2. From the figure, we can see that the majority of clip length lies in the range from 10 seconds to 20 seconds, no matter for what category; the outliners do not exceed 30 seconds.

Regarding sentence complexity, we draw a box plot to see the distribution of caption length among different categories, as shown in Fig. 5.3. st caption lengths are floating around ten words. Although some outliners are extremely long, we can filter them out from the dataset when doing data preprocessing, which won't influence the semantics since each clip is paired with 20 descriptive captions. To conclude, the MSR-VTT dataset is currently the most suitable dataset we could achieve for video diversity, video length, and sentence complexity.

5.1.2. Data Preprocessing

We employ its standard dataset split [171]. It consists of 10000 clips, of which 6,513 are for training, 497 are for validation, and 2,990 are for testing. The ch video is annotated with 20 English captions and a category tag (20 categories in total).

Before we train the model, we prepare the textual data. Firstly, NLTK is used to tokenize words out from sentences as a bag of the words where 'SOS' and 'EOS' are added to the start and end. Words with frequency less than 3 are filtered out. Captions with a length longer than 28 are also removed from the dataset and captions with less than 28 words are zero-padded.

5.2. Metric

We use BLEU@4[109], CIDEr-D[155], METEOR[8], and ROUGE[82] for evaluation. The official codes from Microsoft COCO evaluation server[19] are employed to compute the score. To analyze the readability of captions, we use several readability indexes, which we will introduce in the next chapter.

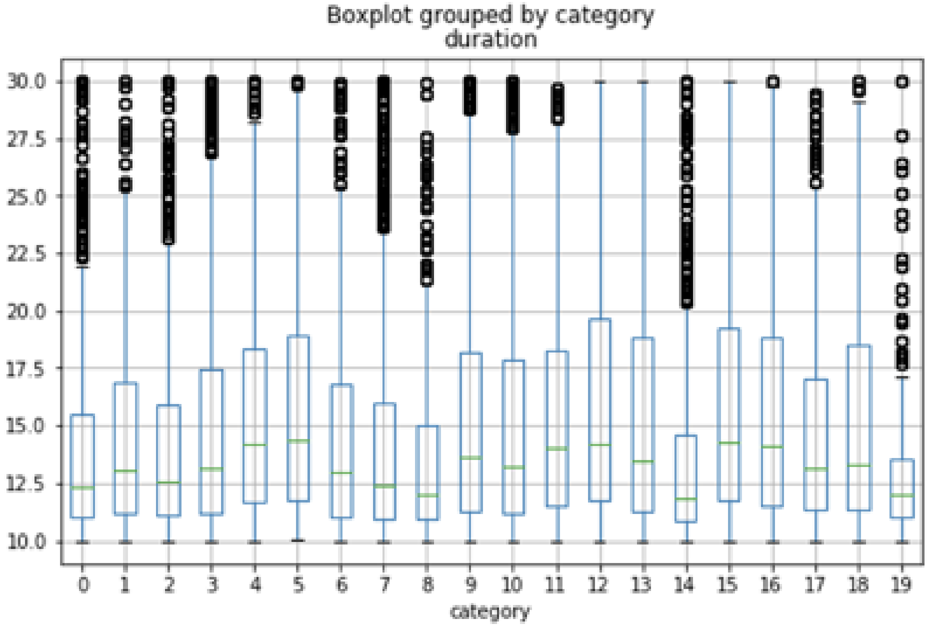


Figure 5.2: The Y axis represents the clip duration (second); the 0 to 19 index in the X axis refers to 20 categories in the sequence of music, people, gaming, sports (actions), news (eventspolitics), education, TV shows, movie, animation, vehicles, how-to, travel, science (technology), animal, kids (family), documentary, food, cooking, beauty (fashion), advertisement, respectively.

5.3. Feature extraction

We use the pretrained Resnet152 provided by **torchvision** [2] to extract visual features. Given a sequence of 16 selected frames, $F = f_1, f_2, ..., f_n$, each frame image is scaled to 224×224 pixel and fed into the pretrained Resnet to get 2048-dimensional feature vector x. As a result, we get a stack of visual features $X = x_1, x_2, ..., x_16$.

We adopt the pretrained 16-frame ResNeXt model that achieves the best accuracy on the Kinetic dataset in the video classification task from [45][3]. Then, we finetune its Conv5x and fc layers on our training set until it converges (25 epochs) and uses the finetuned model to extract short-term dynamics from the videos. Specifically, for each input video, we implemented the spatial transform and the temporal transform. Temporal transform aims to split the video into non-overlapped n 16-frame segments (if the segment is less than 16 frames, we loop itself as many times as necessary). Spatial transform crops every frame image in the segments around a center position to 112x122 pixels. Accordingly, the size of each sample is 3x16x112x112 (channel x frame x pixel x pixel), and the output from the pretrained model for each video is a stack of 2048-dimension features. In the end, we take the arithmetic mean along the stack's axis, resulting in a single 2048-dimension context vector.

We adopted a 16-frame ECO Lite model that pretrained on the Kinetic dataset from [188][4]. Then, we finetune the last fc layers of the 3D net on our training set until it converges (20 epochs) and use the finetuned model to extract long-term action features from the global pool layer of the model. Specifically, for each input video, we transform it 3x16x224x224 (channel x frame x pixel x pixel) and pass it through the pretrained model, which yields a single 512-dimension context vector as the long-term action feature.

We treat the features extracted from pretrained ResNeXt101 and ECO as the context and pad them to the end of frame-level features, which leads to a representation of 16x4608 dimension. In the experiments, we further analyze how the different features influence the model performance.

5.4. Training details

We use negative log-likelihood loss and Adam Optimizer[60]. P torch nn.embedding is employed to take the word token IDs and convert these to word vectors. T e weights for the nn.The embedding layer is learned during the training process as well. The model is optimized by Adam Optimizer[60]. The initial learning rate

[2] https://github.com/pytorch/vision

[3] For details, see: https://github.com/kenshohara/video-classification-3d-cnn-pytorch

[4] For details, see: https://github.com/mzolfaghari/ECO-pytorch

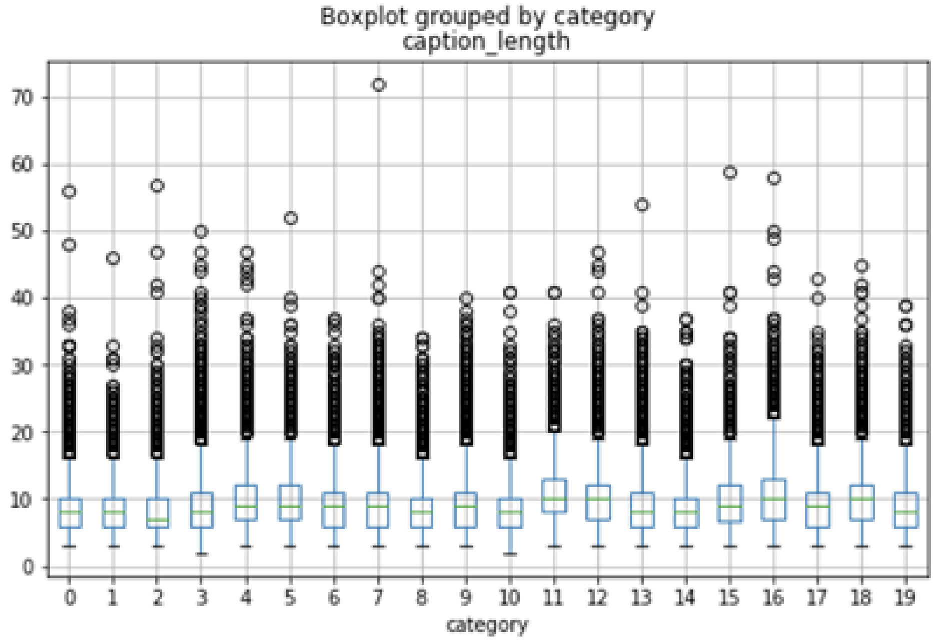


Figure 5.3: Y axis represents the clip caption length; the 0 to 19 index in the X axis refers to 20 categories in the sequence of music, people, gaming, sports (actions), news (eventspolitics), education, TV shows, movie, animation, vehicles, how-to, travel, science (technology), animal, kids (family), documentary, food, cooking, beauty (fashion), advertisement, respectively.

is set to 4e-4 with a weight decay of 5e-4 every every 50 epochs. To avoid an explosion of gradients, we clipped all gradients with five as the maximum norm. The hidden size of the LSTM is 512, and dropout is used with a value of 0.5 for regularisation. A beam search with a size of 2 is used for generating the final captions when testing.

We train the models on a single Titan GeForce GTX GPU with 12GB memory and conduct the following study.

6 Result

The goal of this chapter is to answer **Research Question 3 - 6**. In order to answer these four questions, we carry out several experiments. The chapter is organized as follows:
Firstly, we compare the performance of our model with different add-on components like adjusted attention and various pertained feature models. This subsection is aimed to validate the hypothesis we hold for **Research Question 3**: How can we improve the performance of the baseline video caption model? The next subsection presents the model performance on different categories of video in order to investigate the effectiveness of the solution we give to **Research Question 4**: How can we design the video caption model more sensitive to actions? The following subsection introduces several metrics to measure the readability of our models. It gives evidence that our model can generate sentences that are suitable for people with visual impairment regarding **Research Question 5**: Can the video caption model generate sentences with good readability? In the next subsection, the latency of the model with different add-on components is contrasted. Furthermore, the **Research Question 6**: How to reduce the latency of the video caption model? is addressed. At the end of this chapter, we give some examples of captions generated by our best model against the ground truth.

6.1. Accuracy Analysis

In this subsection, we compare the performance of our model with different add-on components like adjusted attention and various pretrained feature models such as eco and c3d. They are evaluated across four different metrics (Bleu_4, CIDEr, ROUGE_L, and METEOR) on the MSR-VTT dataset. The results are shown in Table 6.1.

	Feature	**Model**	**Bleu_4**	**CIDEr**	**ROUGE_L**	**METEOR**
1	res152 (baseline)	att	36.1	40.1	58.3	26.5
2	res152	adj_att	36.8	41.3	58.9	26.8
3	res+eco	adj_att	38.0	41.7	59.5	27.0
4	res+c3d	adj_att	38.8	42.5	59.4	27.4
5	res+c3d+eco	adj_att	40.0	44.6	60.0	27.7
6	res+c3d(d)+eco	adj_att	38.2	43.1	58.9	26.9
7	res+c3d+eco	adj_att (b=2)	40.2	46.0	60.3	27.9
8	Res101+RN	SGN(b=5)	40.8	49.5	60.8	28.3

Table 6.1: Performance on the MSR−VTT dataset with different add-on components. res152(res), eco, c3d, Res101 and RN denote Resnet152, 16-frame ResNeXt[45],16-frame ECO Lite[188], Resnet101 and 3D−ResNext−101, respectively. They are pretrained models for feature extraction. att, adj_att and b refer to attention, adjusted attention, and beam search width, respectively. SGN[133] is the state−of−the−art model that uses a similar backbone with this work.

With the purpose of seeing if add-on components are effective, the 1st to 7th row in the table shows the results across different metrics. Compared to the other three metrics, CIDEr is more important for evaluating performance. It is because CIDEr is designed for the captioning task, and CIDEr is stated to be more consistent with human judgment than the others[155].

The first row of the table presents the performance of the baseline model. It is an encoder-decoder architecture with a soft attention mechanism. Pre-trained resnet152 is used for its feature extraction. When adjusted attention layer is added to the baseline model, there is a performance increase on all four metrics, especially on the CIDEr metric from 40.1 to 41.3. This indicates that the adjusted attention layer helps to improve the performance of the baseline model.

From the 2nd to 5th row in the table, we study the add-on features (res152, eco, and c3d feature). The accuracy of the model with adjusted attention and resnet152 feature is presented in the 2ed row, 36.8, 41.3,58.9, and 26.8 in Bleu_4, CIDEr, ROUGE_L, and METEOR, respectively. By adding the eco feature or c3d feature, the performance of the model is improved across all four metrics. This implies that these two features give the model more dynamic information that resnet152 can not provide to generate the correct sentences. The most obvious enhancement is on the Bleu_4 metric. The eco feature and c3d feature help the model improve from 36.8 to 38 and 38.8 separately. This may because Bleu_4 is initially designed for assessing documents, these two features assist the model produce longer and richer sentence.

If eco and c3d features are employed together (Row 5), the model with an adjusted attention layer gets a more significant boost in performance. Compared to the model only using the resnet152 feature, there is an 8.7%, 8.0%,1.7%, and 3.4% increase in Bleu_4, CIDEr, ROUGE_L, and METEOR, respectively. This pander to our assumption that eco and c3d can provide different kinds of action features. The eco feature is for long-term action, while c3d is for short-term dynamics. They are complementary in capturing action in videos. Since the c3d pretrained model takes much time to extract features from the video, we do down−sampling on video frames to see how the performance alters. Row 6 in the table shows that down-sampling will not decline the performance much. As CIDER is most consistent with human judgment, the score is only dropping from 44.6 to 43.1.

On top of it, the performances of model on CIDEr can be further improved to 46 if beam search technique with beam size 2 is applied (Row 7). Compared to the baseline model, Our model has a 14.7% increase on CIDEr.

6.2. Category Analysis

In this section, we evaluate the performance of each model in different video categories across different metrics. To see if adding eco and c3d features helps the model be more sensitive to actions.
There are 20 video categories in total, including music, people, gaming, sports/actions, news, education, TV shows, movies, animation, vehicles, how-to, travel, science (technology), animal, kids (family), documentary, food, cooking, beauty (fashion), advertisement. Four different metrics, Bleu_4, CIDEr, ROUGE_L, and METEOR, are used to evaluate the performance. Three models are compared here: a model with resnet, a model with resnet and eco, and a model with resnet, eco, and c3d.

In the following sections, r, c, and cc are used to represent the model with resnet, a model with resnet and eco, and a model with resnet, eco, and c3d, respectively. In the following four line charts, we introduce three new representations, c_over_r, cc_over_r, cc_over_c. c_over_r represents the percentage of improvement in the model with resnet, and eco gets over the model with resnet. The same goes for cc_over_r and cc_over_c. We first see the performance of each model in all video categories generally. And then, we look at the howto and sports/actions categories since these two categories contain more complex Human actions.

6.2.1. Improvement on Metric CIDEr

For the line charts below, we can see the model improvement on metric CIDER.

In the first line chart, Model c has a positive enhancement over Model r in 12 categories. The greatest one is on tv-show which is over 20%. In the second line chart, Model cc has an positive enhancement over Model r in 12 categories, with the greatest one exceeding 30% on tv-show. Comparing model c and model cc, we can see the third line chart illustrates that Model cc defeats Model c in tv-show and beauty by a large margin while witnessing an evident decrease in education. We look at the last line chart; model cc has a small advantage over Model c. However, Model cc only has a performance rise in half of the categories. The margin of enhancement is still larger than that of the decrease in the other half of the categories.

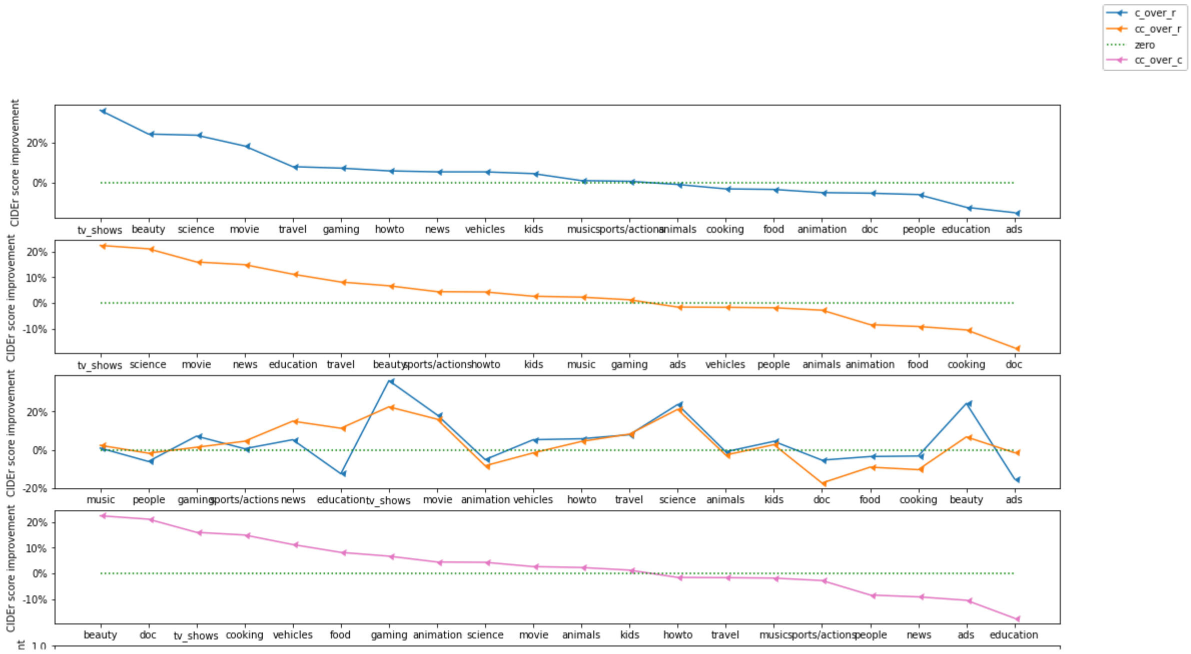


Figure 6.1: Improvement on Metric CIDEr

To conclude, in terms of CIDER, Model c and Model cc outperform Model r in majority of video categories. Both Model c and Model cc has an improvement on howto and sports/actions categories compared to Model r. Model cc takes advantage over Model c in most categories, but the improvements are not significant. Besides, Model c performs slightly worse than Model c in the howto and sports/actions categories.

6.2.2. Improvement on Metric METEOR

For the line charts below, we can see the model improvement on metric METEOR.

In the first line chart, Model c has a marginal enhancement over Model r in 13 categories. The one with the highest rise is in how-to, which is 5%. There is an obvious decline in ads, bottoming out to 8%. In the second line chart, Model cc has a positive enhancement over Model r in 14 categories, with the greatest one exceeding 8% on beauty. Similar to Model c, Model cc has decline in ads by 8% as well. Comparing Model c and model cc, the third line chart illustrates that Model cc defeats Model c in tv-show and beauty by large margin while witness the evident decrease in education. Looking at the last line chart, Model cc has an evident advantage over Model c. Model cc only has small performance decline in 4 categories and a 8% drop on education.

In summary, in terms of METEOR, Model c and Model cc still outperform Model r in majority of video categories. Both Model c and Model cc has an improvement on howto category compared to Model r but the performance on the sports/actions category is marginally enhanced. Model cc take evident advantage over Model c in most of categories but have no enhancement on the sports/actions category.

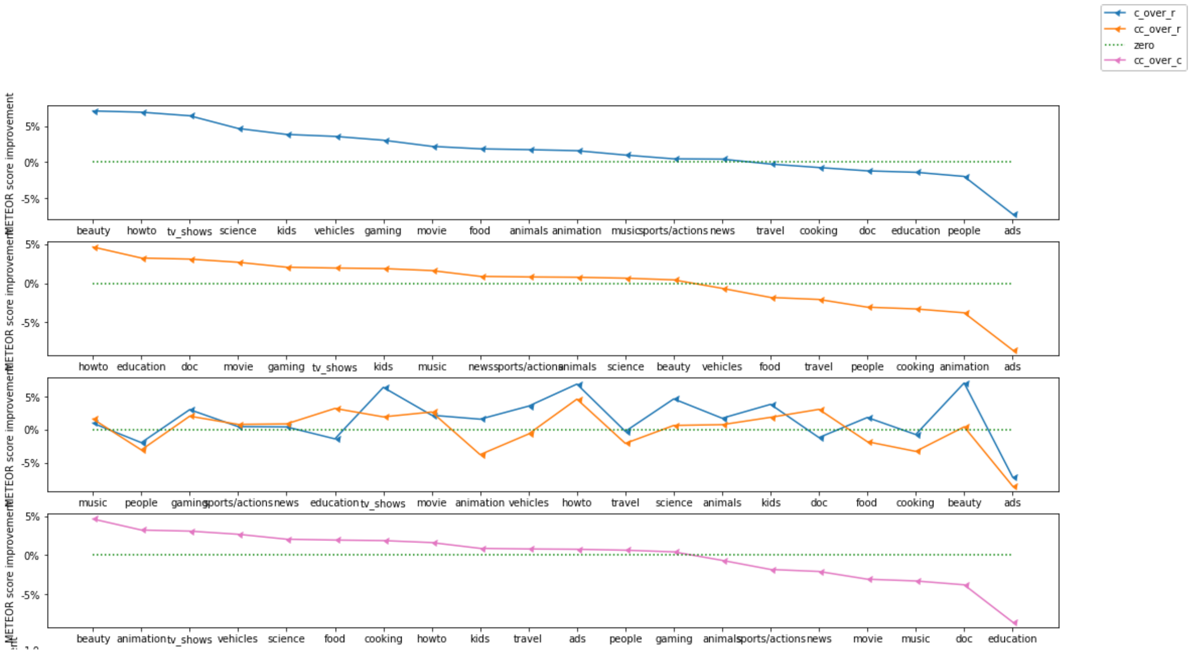


Figure 6.2: Improvement on Metric METEOR

6.2.3. Improvement on Metric ROUGE_L

For the line charts above, we can see the model improvement on metric ROUGE_L.

In the first line chart, Model c has a marginal enhancement over Model r in 13 categories. The one with the highest rise is in tv-show which is 5%. There is a decline in ads, exceeding 4%. In the second line chart, Model cc has a positive enhancement over Model r in 14 categories. Three categories (beauty, howto and tv-show) have a surge above 5%. While Model cc still has decline in ads by 3%.

Comparing Model c and model cc, we can see the third line chart illustrates that Model cc defeats Model c in animals, animation, and vehicles by a large margin while witnessing a slight decrease in education by 2%. Looking at the last line chart, we can see model cc has an evident advantage over Model c. Model cc only has small performance decline in 3 categories and a 3% drop on education.
To conclude, in terms of ROUGE_L, Model c and Model cc still outperform Model r in majority of video categories. Both Model c and Model cc has an improvement on howto category compared to Model r but the performance on the sports/actions category is slightly enhanced. Model cc take evident advantage over Model c in most of categories but smaller enhancement on on the howto and sports/actions categories.

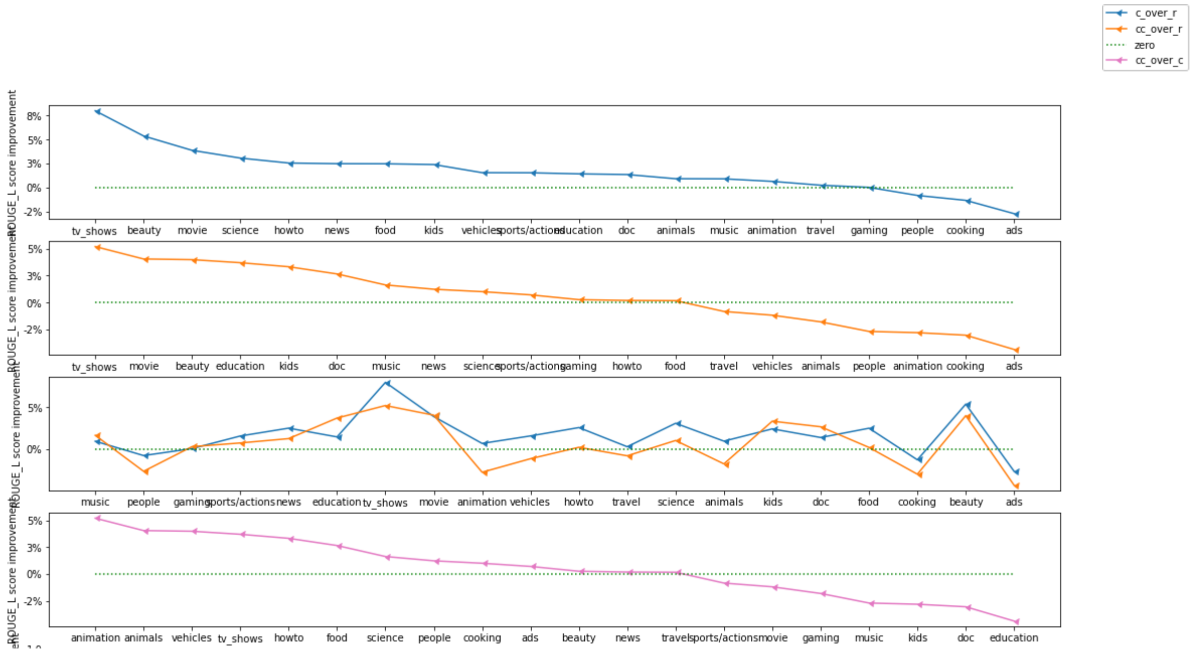


Figure 6.3: Improvement on Metric ROUGE_L

6.2.4. Improvement on Metric Bleu_4

For the line charts above, we can see the model improvement on metric Bleu_4.

In the first line chart, Model c has an enhancement over Model r in 12 categories. The one with the highest rise is movie which is nearly 15%. There is a large decline in ads, around 15%. In the second line chart, Model cc has a considerable enhancement over Model r in 15 categories. Three categories (movie, science and tv-show) have a surge above 10%. While Model cc still has a decline in ads by more than 10

Comparing Model c and model cc, we can see the third line chart illustrates that Model cc defeats Model c in tv-show and animation by a large margin while witnessing an evident decrease in movie. Looking at the last line chart, we can see model cc still has an evident advantage over Model c. Model cc only has performance decline in 2 categories, around 10% drop on people and movie.

To conclude, in terms of Bleu_4, Model c and Model cc outperform Model r in majority of video categories. However Model c and Model cc do not have an improvement on the howto and sports/actions categories. Model cc take evident advantage over Model c in most of categories. And Model c and model cc have the similar performance on these two categories.

6.2.5. Conclusion

- Adding eco and c3d features helps the model to perform better on many video categories across four metrics. As the MSRVTT dataset is rich in action among categories, it has been widely used for action recognition tasks. From this point on, eco and c3d features help the model improve. This implies that they enhance the sensitivity of the model to the actions.
- The model performance on the howto and sports/actions categories is only slightly improved. This may be because the actions contained in videos from these two categories are more complex. The sports/actions category contains a lot of sports videos that include many people, which adds to the difficulty for the model in capturing the most action and generating a single sentence. Videos in the howto category are more trivial; most of them are instructions that teach people to do something step by step and normally with audio. This enhances the challenge for the model to summarise.
- The model performance drop on ads and education is predictable for two reasons. The videos related to ads and education are always accompanied by rich audio as supplementary information. We exclude the audio of the dataset from training. Ads and education videos generally involve more stationary objects like products and text. Our model is designed to be tailored to human action. Thus, the model

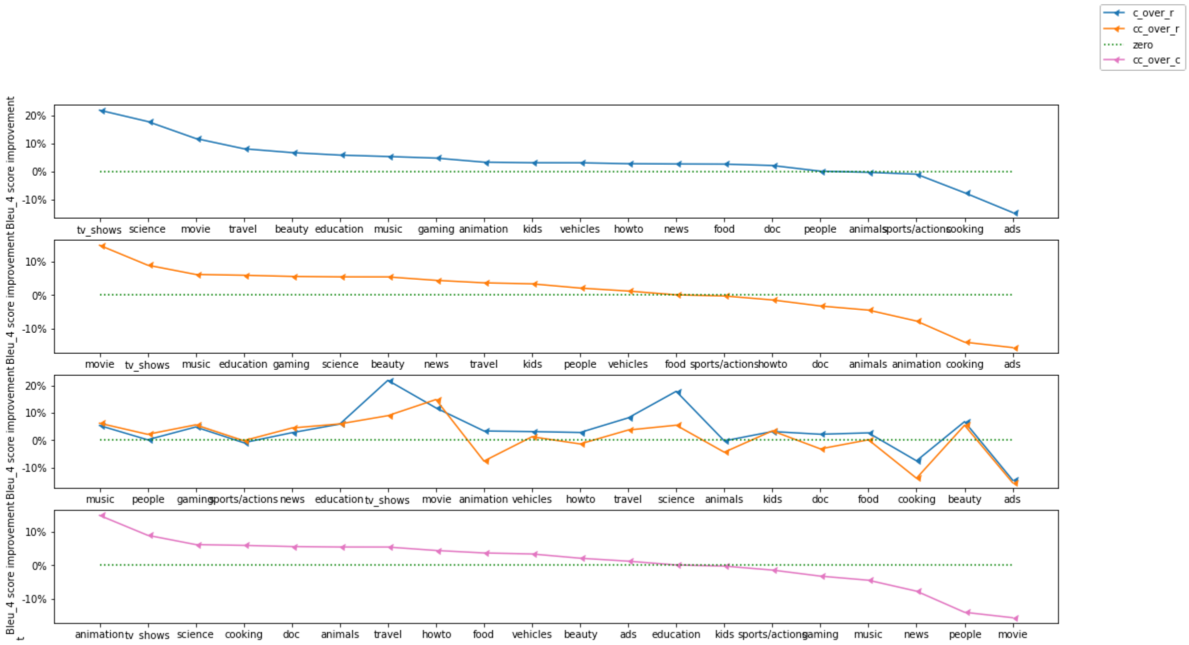


Figure 6.4: Improvement on Metric Bleu_4

may be less sensitive to ads and educational videos.

6.3. Readability Analysis

From the User Study 3, we know that visually impaired people prefer generated sentences by video caption models that are easy and simple to understand. This results in our **Research Question 5: Can the video caption model generate sentences with good readability?** In addition, we add an extra adjusted gate to the attention mechanism in the model, which we intuitionally believe can guide the decoder to learn a better attention weight. Thus, the model can generate sentences with better readability.

As to answer this research question and check the effectiveness of adjusted gate, we conduct a study. e use several readability metrics to evaluate sentences generated by the models with and without adjusted gate. eadability score is a computational index normally based on sentence length, syllable density and word familiarity. t can tell us the age range and education level the reader need to comprehend the input text.

6.3.1. *Experiment setup*

We choose the model with attention mechanism (the baseline model) and the model with adjusted attention mechanism (the second model refers to Accuracy table 6.1) as a comparison. e only select these two models for two reasons. e would like to validate if an extra adjusted gate can guide the model generate sentence with better readability. We also want to exclude the interference caused by the pretrained model from the result. There is very rare literature on the intersection of readability and people with visual challenges. owever, this research [125] implicates that the use of short words and the including more familiar words can improve readability and comprehension for people with dyslexia. This finding is inspiring. The SMOG Index, The Gunning Fog Index, and Flesh Reading Ease involve syllables counting in computation, though in different ways. That means these three metrics favor text with short words. When it comes to including more familiar words, The New Dale Chall can be helpful since it uses a set of predefined common words to evaluate. n addition, the New Dale Chall is reported to be more suitable for student[98], which exactly coincides with our target audience. Therefore, we select these four metrics to evaluate the readability of sentences from our models. Since readability metrics are designed to evaluate text. pply them in every single sentence will lead to bias. e aggregate all captions generated for test set and treate them as a whole text input to four different metrics.

6.3.2. *Result*

Table 6.2 presents the readability scores across four selected metrics. For simplicity, we refer Model att to the model with an attention mechanism and model adj_att to the model with an adjusted attention mechanism. enerally, two models get quite similar performance across four metrics. Model att gets 5.68, 3.85, 85.39, and 4.90, scored by SMOG, Gunning Fog, Flesch Reading Ease, and Dale Chall, respectively; Model adj_att gets 5.46, 3.80, 85.97, and 4.94 scored by SMOG, Gunning Fog, Flesch Reading Ease, and Dale Chall respectively. inIncewo models differ marginally in scores by each metric; the recommended reader age falls in the same range: 11-14, 7-11, 12-13, 10-12 suggested by SOMG, Gunning Fog, Flesch Reading Ease, and Dale Chall in the order given.

Querying the Interpretation Table of four metrics (table 2.3, 2.4 and 2.6) given in section 2.5, we can see that two models are evaluated to be easily read by the public. The suggested age range of 7-14 is also below the age of our target audience. Model adj_att outperforms Model att in SMOG and Gunning Fog by 3.9% and 1.3% while is defeated slightly in Flesch Reading Ease and Dale Chall by 0.68% and 0.82%. Since SMOG and Gunning Fog favor text involving fewer syllables, this result implies that Model adj_att probably can generate sentences involving more short words compared to Model att. the decline in Dale Chall indicates that Model adj_att may tend to generate sentences including words that are unfamiliar to 4th-grade students. However, from the writer's perspective, regarding the significant accuracy improvement (by 3% in CIDEr), the decline in Flesch Reading Ease and Dale Chall is too marginal to reflect the tendency.

To answer **Research Question 5**, the video caption model can generate sentence with good readability and it is suitable for our target audience and the public to read easily. y adding an extra adjusted gate to attention mechanism, it increases the model performance in accuracy without adding difficulty in reading.

Table 6.2: Readability Score across four metrics
att represents the model with attention mechanism; adj_att represents the model with an adjusted attention mechanism.

Model		**Smog**	**Gunning fog**	**Flesch**	**Dale Chall**
att	score	5.68	3.85	85.39	4.90
	age range	11-14	7-11	12-13	10-12
adj_att	score	5.46	3.80	85.97	4.94
	age range	11-14	7-11	12-13	10-12
Interpretation		ideal for average readers	very easy to read	fairly easy to read	easy to read
Improvement		3.9%	1.3%	-0.68%	-0.82%

6.4. Latency Analysis

In this subsection, the latency of the model with different add-on components is contrasted. Then, we try to analyze the relation between video and latency in terms of video length and the number of frames. Furthermore, we try to find a proper answer to the **Research Question 6**: How to reduce the latency of the video caption model? By downsampling and changing the model parameters, the latency of the model has been reduced.

Firstly, we plot two histograms to see the distribution of video length and video's number of frames separately in the test dataset, as shown above. The left histogram for the video length presents a long-tail like distribution. All the video are less than 30 seconds. About 65% of video is short than 15 seconds and only 5% video are longer than 25 seconds. The right histogram shows nearly 60% videos have number of frames in the range from 250 to 400.

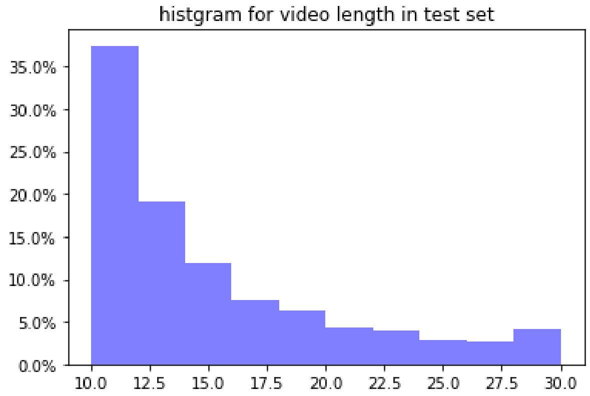


Figure 6.5: Histgram for video length in the test set.

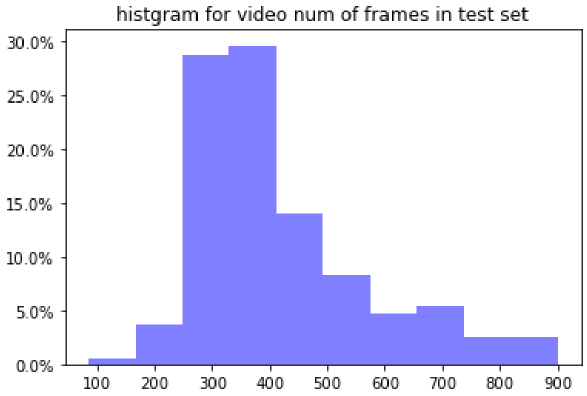


Figure 6.6: Histgram for video's num of frames in test set

6.4.1. Latency

At the beginning of this study, the model had a high latency, but then we found out that it was because of a conflict between Pytorch and Windows. An error was caused by setting num_workers, which is the parameter of Dataloader in Pytorch, greater than 0 in the Windows environment.

When the value is 0, only the primary process is used to load the dataset, and when the value is greater than 0, the corresponding number of sub-processes will be created to be responsible for the related data loading work. This method uses the multiprocessing package to create a new subprocess and multi-process synchronization, and the fork function can be used to execute a new subprocess from the corresponding code (using the fork statement) in Linux-related systems, while there is no corresponding function in Windows-related systems. This difference will lead to some code being executed repeatedly in the relevant training code in the Windows environment, resulting in some exception errors. Thus, once we set the num_workers to zero, the model latency meets our requirements.

Latency histogram
The latency of the model with different add-on components is shown in Fig. 6.7.

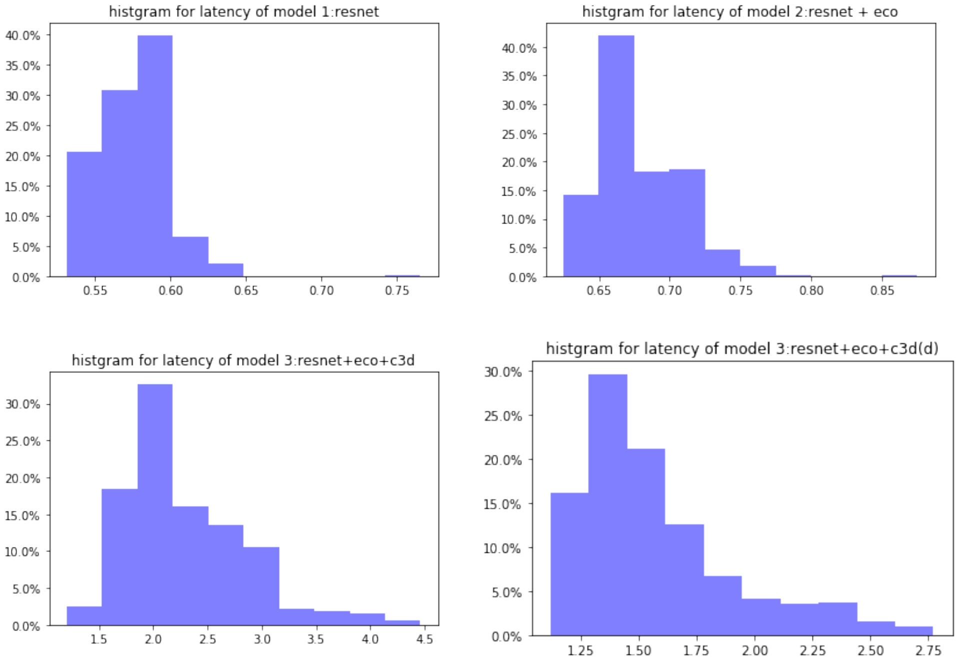


Figure 6.7: Lantency distribution

From the figures, we can clearly see that:

- Model 1 is the model with adjusted attention model, which is only trained on the reset feature. The latency on all the test videos is less than 0.65 seconds.
- Model 2 is the model with adjusted attention model, which is trained on resnet and eco features. Compared to Model 1, the latency has increased a little. The latency of the majority is below 0.8 seconds.
- Model 3 is the model with adjusted attention model, which is trained on three features: resnet, eco, and c3d. With the joining of c3d, the latency has significantly increased. The latency on 90% of videos is less than 3 seconds. There are some extreme cases whose latency exceeds 4 seconds. The outliers are further analyzed in the next section.
- Model 3(d) is the model with adjusted attention model, which is trained on three features: resnet, eco, and c3d. We have witnessed a large-time increase by using the c3d model; we evenly downsample the video to 24 frames per second. Compared to Model 3, the latency has been reduced a lot. The latency of the majority is below 2.75 seconds.

Latency vs. video length and number of frames
To explore the the relation between the relation between latency and video in terms of length and number of frames, we plot line charts for each model. From the figures 6.8,6.9,6.10 and6.11 , we can clearly see that:

- Generally, the latency of model 1 fluctuates around 0.5 seconds with a margin between 0.65 and 0.25 seconds on both the latencyframe and latencyduration plots. When the duration is close to 30 seconds, and the number of frames reaches 900, the latency jumps to 0.75 seconds.
- The latency of model 2 has a similar trend with model 1 but has increased a little. It fluctuates in the range from 0.2 to 0.8 seconds, with a mean of around 0.575 seconds. It rises dramatically over 0.85 seconds when the duration is close to 30 seconds, and the number of frames reaches 900 as well.
- The latency of model 3 and model 3d share a similar trend. The relation between latency and frame number or duration is approximately linear.

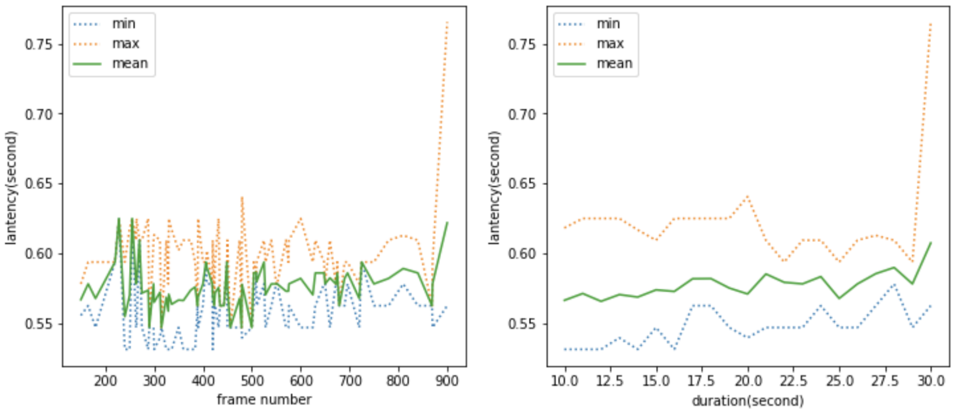


Figure 6.8: model 1: resnet

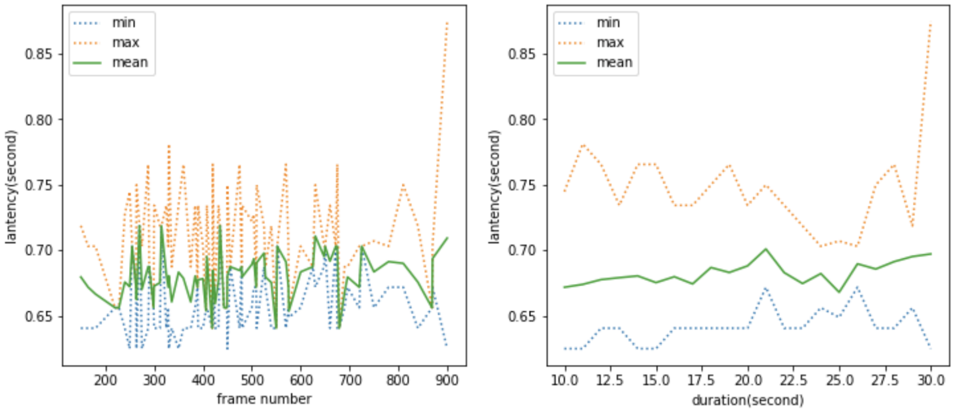


Figure 6.9: model 2: resnet+eco

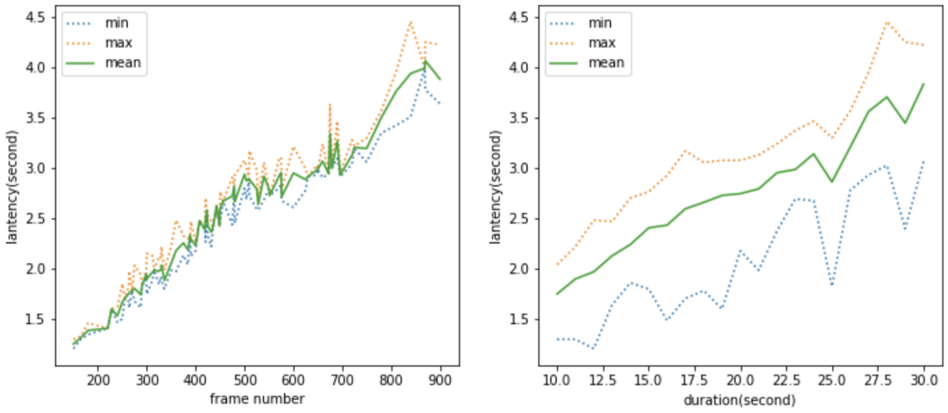


Figure 6.10: model 3: resnet+eco+c3d

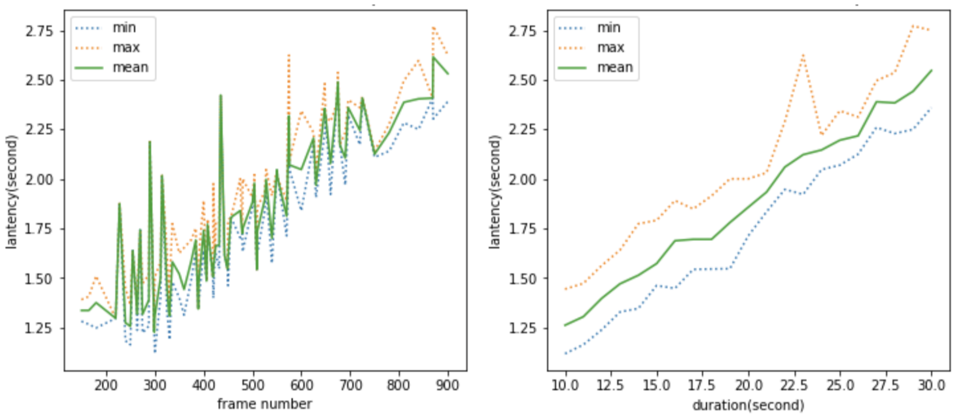


Figure 6.11: model 4: resnet+eco+c3d(downsample)

6.4.2. Comparison

In this section, further analysis on outliers and is conducted. What leads model 3 to be timeconsuming is investigated as well.

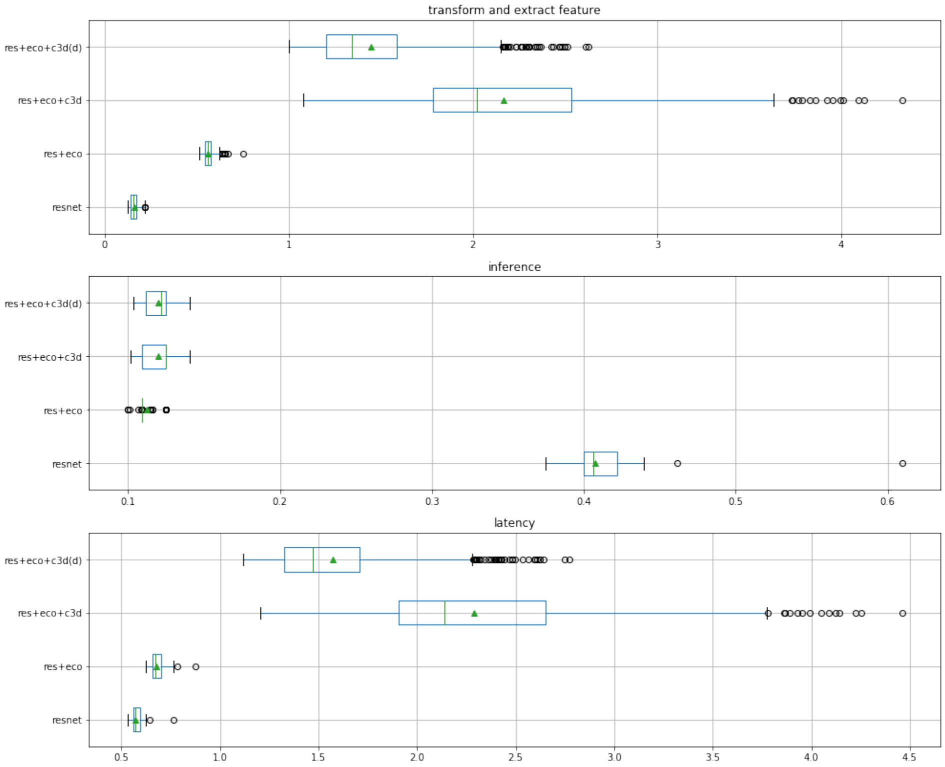
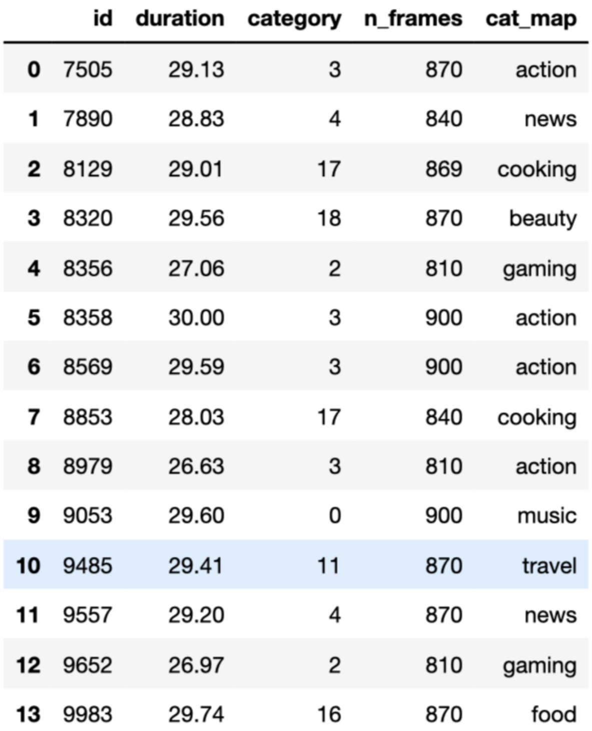


Figure 6.12: Boxplot

We divide the whole process into two parts: transform and extract feature inference. Transform and extract feature is the time used to prepare the features; inference is the time used to generate sentences. The time used by each part is measured. The result is presented in three box plots; see Fig. 6.12. Firstly, we look into the bottom box plot. Resnet (Model 1) and resnet+eco (Model 2) have very low latency and are distributed tightly within a range from 0.5 to 1.0 seconds. Regarding resne+eco+c3d (Model 3), it is comparatively widespread, with a mean of around 2.25 seconds. While it has many outliers. Similarly, resnet+eco+c3d(d) (Model 3 with downsampling) has many outliers as well, and its mean is around 1.5 seconds. Comparing the first and second box plots, we can infer that the transform and extract feature contributes the most time to the latency. Also, the outliers are from this part as well with respect to Model 3 and Model 3d.

Outliers
From the box plots and histograms, we can see there is some outliers. In this part, those videos whose latency exceed the third quantiles are defined as the outliers. These video are drawn from the dataset and their information are listed in below table.

There are a total of 14 outliers, as shown in the table. It is the video's original ID in the MSRVTT dataset, where all the videos have the exact resolution. Duration stands for the video length in seconds. n_frame is the total number of frames contained by the video. Category or cat_map represented the video category, as the name implies. All the outliers have long lengths, ranging from 26 to 30 seconds roughly. Their number of frames is more than 800. It is interesting to find that 5 out of 14 outliers are from the sports/actions category. This is to exclude the possibility that videos in the sports/actions category are much longer than videos in other categories. We check the mean of n_frame of videos in different categories. The mean varies from 330 to 450 frames, and 15 out of 20 categories have a mean of n_frame more than 400 frames. The means of n_frame of videos in sports/actions category is 432. It does go above average. That suggests that the videos from this category contains richer information in frame than others.



	id	duration	category	n_frames	cat_map
0	7505	29.13	3	870	action
1	7890	28.83	4	840	news
2	8129	29.01	17	869	cooking
3	8320	29.56	18	870	beauty
4	8356	27.06	2	810	gaming
5	8358	30.00	3	900	action
6	8569	29.59	3	900	action
7	8853	28.03	17	840	cooking
8	8979	26.63	3	810	action
9	9053	29.60	0	900	music
10	9485	29.41	11	870	travel
11	9557	29.20	4	870	news
12	9652	26.97	2	810	gaming
13	9983	29.74	16	870	food

Figure 6.13: Outliers
outliers of Model 3d and Model 3d. It is the video's original ID in the MSRVTT dataset, where all the videos have the same resolution. Duration stands for the video length in seconds. n_frame is the total number of frames contained by the video. Category or cat_map represented the video category, as the name implies.

We randomly draw one outlier from the sports/actions for the case study. Video7505 is selected, which is 29 seconds long and 870 frames. By inspecting this video (see figure 6.14), we find the video content is considerably rich: a man is getting interviewed and is talking about the accident that happened to him in the sea. Then, the video shows how he was rescued by people in the speedboat. At last, the man shows up again, touching his head with his hands. In summary, these three factors add to the increase in latency.

From the table 6.13, we can see all the outliners are longer than 26 seconds and contain more than 800 frames. In contrast, we randomly drew one video from the test set, which met the same conditions but was not an outliner. Video7339, which is 27 seconds long and 810 frames. However, the video is long and has a large number of frames. Its content is comparatively unitary (see figure 6.15): A firework blooms and fades away slowly in the night sky, accompanied by the cheers of the people in the background.

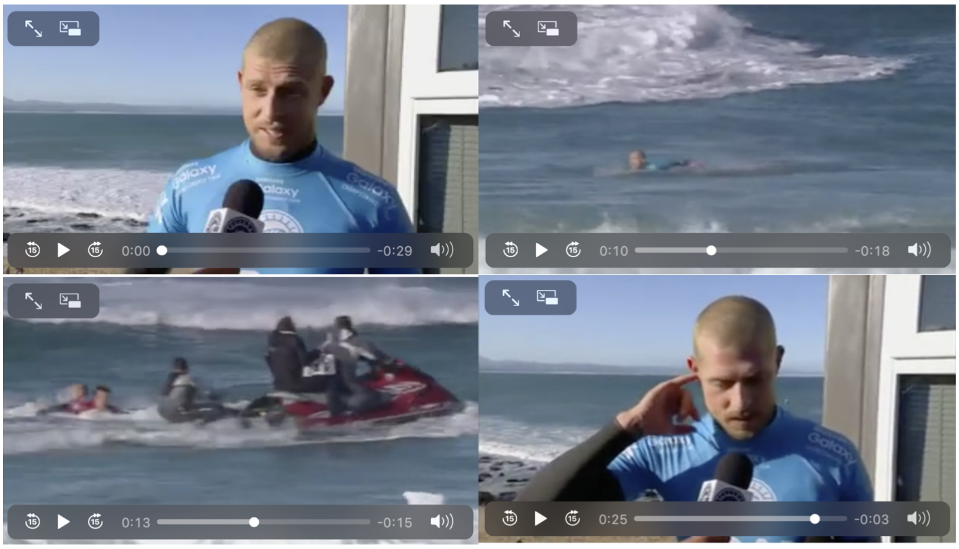


Figure 6.14: Video7505

A man is getting interviewed and is talking about the accident that happened to him in the sea. Then, the video shows how he was rescued by people in the speedboat. At last, the man shows up again, touching his head with his hands.

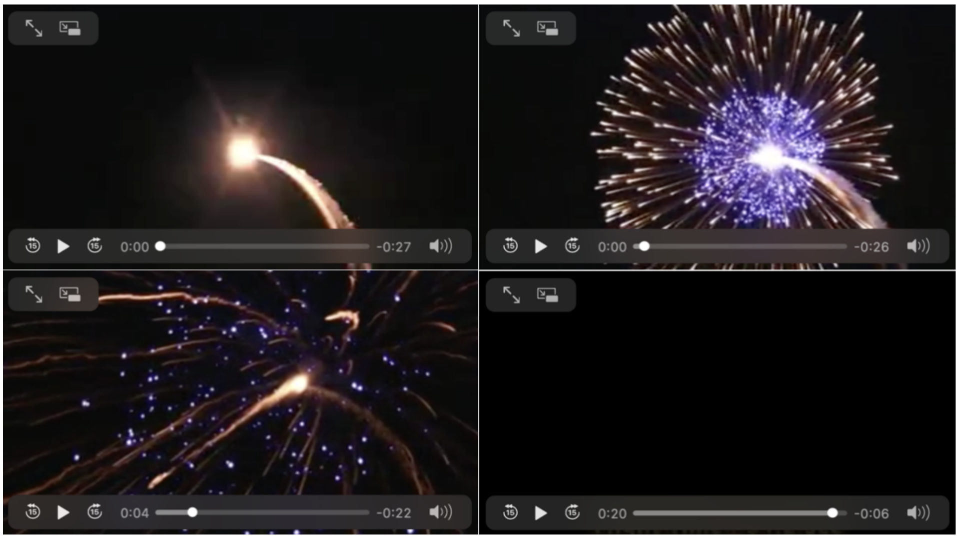


Figure 6.15: Video7339

A firework blooms and fades away slowly in the night sky, accompanied by the cheers of the people in the background.

6.5. Case study

This subsection presents some good or bad examples of captions generated by our best model(adjusted model with resnet, eco, and c3d features) against the ground truth.

6.5.1. Good examples

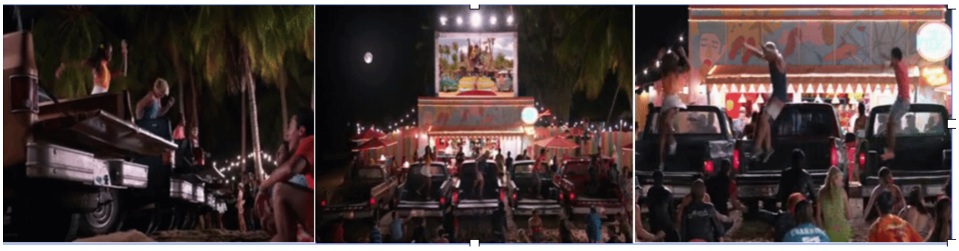


Figure 6.16: Good example 1

GT: a crowd of people are dancing. Ours: a group of people are dancing

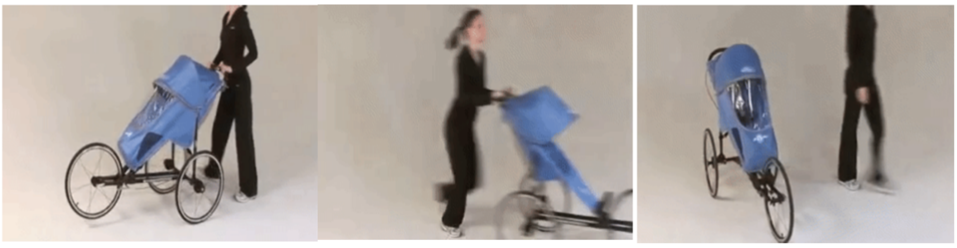


Figure 6.17: Good example 2

GT: an advertisement about a baby stroller. Ours: A woman is showing us how to use a stroller.

- Good example 1. Although our model uses another phrase, a group of ', to substitute a crowd of people,' the caption generated expresses the same meaning.
- Good example 2. Two captions have different biases. The ground truth explains the scene where this video takes place. Our captions directly describe the activity contained in the video. In another way, it proves that our model is more sensitive to the action in the video.

6.5.2. Bad examples

- Bad example 1. Comparing the ground truth and caption generated by our model, our model failed to recognize it as a ‘new’ system in the video game. There are probably two reasons behind it. The model wasn’t specifically trained to recognize the text in the image. We didn’t use the audio to train the model.
- Bad example 2. Our model succeeded in detecting the activity but failed to recognize the color dress of the dominant woman in the video. It may be interfered with by another model in a black dress in the background.

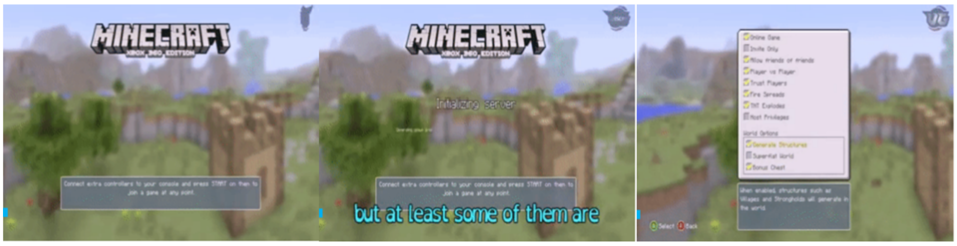


Figure 6.18: Bad example 1

GT: there is a man talking about a new system Ours: a person is explaining about the video game

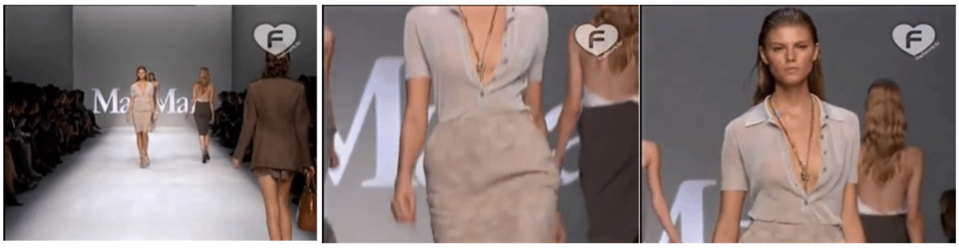


Figure 6.19: Bad example 2

GT: a female model is walking down a runway. Ours: a woman in a black dress is walking down the runway

7

Discussion

In this chapter, the Limitation of this work will be discussed and the direction of furture work will be also pointed out.

7.1. Limitation

Throughout this thesis, some limitations will be discussed. It is worth to analyze and understand not only for this thesis but also for future work. We will address the limitations in four aspects as below.

- Sampling. In this thesis, we are subject to the pre-trained models (Resnet and ECO) used for feature extraction, and we sample a fixed number of frames from videos. With respect to another pretrained C3d model, we use all frames for feature extraction. From the ablation study on latency, we see that the bottleneck of latency lies in the feature extraction, and the time consumed by the C3d model among the three pre-trained models is dominant. By downsampling at a fixed frame rate for the C3d model, the latency has been significantly reduced. Meanwhile, it still does not meet the latency requirement of our target audience. A dynamic sampling strategy is likely to be ideal for practice.
- Latency. As mentioned in this work, the model with the best performance does not meet the latency requirements, and its bottleneck relies on feature extraction by using the C3d model. And the latency can be reduced by changing the sampling strategy. Currently, the model is running on a single Titan GeForce GTX GPU with 12GB of memory. If we have multiple GPUs, we can use data parallel or model parallel with pipeline input to speed up according to Pytorch Documentation [1]. In addition, we can also assign each pre-trained model to each GPU to extract features simultaneously to reduce latency. [78] Another possible solution is to use the quantization technique to scale down the model itself.
- Human Evaluation. Apart from automated evaluation metrics, human evaluation is also often used to judge the quality of machine-generated captions as a supplement. In this thesis presented above, we have not covered this. We have considered and designed a human evaluation survey, but our implementation of human evaluation has been hindered by several obstacles. We have designed a survey on the Online crowdsourcing platform Figure eight [2] where 20 most representative videos are enclosed together with captions generated by different models. We employ a tutor (as our participants are students from Visio, one of them is the best candidate who understands this project but also knows the visually impaired people very well) to explain the purpose of the survey and guide the participants all the time. The tutor's main responsibility is to describe the video content from the perspective of the visually impaired people to our participants. Then, with the description given, the participants are required to score several models in terms of three measurements: Correctness, Complexity, and Relevance. In addition, we asked the coach to write down the description beforehand to maintain consistency. Accordingly, we can find the best model. However, the result is unsatisfying. We see homogeneous scores across these three measurements per model. This is obviously not what we want since it indicates that Correctness, Complexity, and Relevance are probably not distinguishing criteria for visually impaired

[1] https://pytorch.org/tutorials/intermediate/model_parallel_tutorial.html

[2] https://f8federal.com

people to evaluate captions. From the first iteration of the human evaluation survey, we also learned that our participants had difficulty accessing the online crowdsourcing platform and understanding the survey. Without a tutor who knows the project and participants well to guide the process all the time, we probably cannot collect valuable and authentic results. This leads to the conclusion that the survey can not be conducted online. In the second iteration of the human evaluation survey, we decided to use usefulness as the measurement instead. The participants are supposed to rank the captions given. Usefulness is defined as the subjective score, with highest rank to the "Most useful" and lowest rank to the "Least useful". No two captions should be given the same rank unless they are identical. The second iteration of human evaluation survey was suspended firstly due to scheduling problem with tutor and then has been aborted due to pandemic.

- Dataset. There is no available video dataset that is annotated with descriptions for visually impaired people. Besides, due to the pandemic and privacy issues, it is not possible to collect data within a short time. The MSR-VTT dataset is the most suitable one for meeting the requirements of visually impaired people, although there exists some ambiguity in the division of categories. If we could collect the data directly from people with visual challenges, it would be helpful to train a model that fits their needs better.

7.2. Future work

This thesis has presented the preliminary work on video captioning for the visually impaired people. For future work, there are several directions can be further investigated.

More work can be done on the sampling strategy. As we discussed in section Limitation, a dynamic sampling strategy would be more practical: Sample more frames for longer videos while sampling fewer frames for shorter videos. It would also be interesting to investigate the informative frame picking in the video, sampling frames that contain more information for video captioning.

With respect to reduce latency, quantisation technique can be performed to scale down the complexity of model. By storing tensor at lower bitwidths and etc., the model can be more compact and can be deployed on hardware platforms easier.

More effort can be devoted to human evaluation. Human evaluation plays a crucial role in natural language processing. However, there are no perfect experimental and reporting standards for the non-blind community[153], let alone the visually impaired people. More research on the visually impaired community can be helpful to create a more sensible evaluation criterion for them. If human evaluation can be carried out, the result will give the direction to enhance the current model.

Currently, there is no available video dataset that annotated with description for the visually impaired people particularly. If such dataset can be created, it would a great contribution in the research area of assistive technologies, computer vision, nature language processing (NLP) and even crowdsourcing.

8 Conclusion

This chapter will answer the following research questions set out in this thesis:

How to develop a video captioning model for the visually impaired?

This main question can be split into six sub-questions, and these questions are answered below.

1. What techniques can be used to build a video caption model for the visually impaired?
2. What are the requirements for video captioning special from the visual impaired?
3. How can we improve the performance of the baseline video caption model?
4. How can we design the video caption model to be more sensitive to actions?
5. How can the video caption model generate a sentence with good readability?
6. How to reduce the latency of the video caption model?

Question 1: In order to solve the first research question, we start with a literature review to investigate the contemporary state-of-the-art of video captioning and applications for the visually impaired in the related field. In this literature review, firstly, we give a brief introduction in section 2.1 on video caption as well as an explanation of some terminologies used in the field. Next, we dive into the pool of different video caption methods from the perspective of template-based generation, seq2seq model, reinforce learning, adversarial learning, and the latest works that employ transformer in section 2.2. In section 2.3, we present relevant video datasets in the area of video understanding and highlight ones that are significant to the video captioning task. Later, existing evaluation and readability metrics are reviewed and further discussed in terms of their merits and drawbacks in sections 2.4 and 2.5. In the coming section 2.6, we show the practical application for the visually impaired in the field of visual understanding.

To conclude, recently, encoder-decoder architecture has become the main-streaming backbone of video caption models with different deep learning techniques like attention, reinforcement learning, and adversarial learning as the previous template-based method lacks the scalability with the dramatic increase in data amount.

Question 2: In order to answer this question, we conducted a survey with the people in Visio twice. The first questionnaire is designed to get basic information about the visually impaired people at Visio and their dilemmas in life. The second questionnaire is designed based on the information collected from the first one. With the purpose of further verifying the needs of the target audience, we propose a scenario: There is an app on the mobile phone. They can use it to get some information about the environment they are in or things that happen nearby by taking a video of their surroundings (the algorithm behind the app is what we aim at). From this survey, we get some requirements that help us design the video captioning model, as shown below.

1. The video should not be longer than 30 seconds.
2. The Latency should be limited to 3-4s.
3. Scene: when there is loud music/noise

4. The dataset should cover a wide variety of topics. (People, sports/actions, vehicles/autos, howto, travel, animals/pets, kids/family, food/drink, cooking, beauty/fashion, advertisement)
5. The video should be described in a single sentence briefly.

Question 3: Considering the user requirements and the pursuit of performance together, we put some effort into enhancing the model accuracy from 3 aspects:

1. Video representation. The video is fed into three different pre-trained dens (Resnet, ECO, and Res3D) for features of different levels(frame-level feature, long-term action feature, and short-term dynamics), ultimately yielding a representation of the video by concatenation. Particularly, we tailor our design to lay more emphasis on capturing video dynamics since we know that the visually impaired care more about the actions in the video content from a user study.
2. A modified attention layer. Complementary to the video representation obtained from the pre-trained model, the attention layer is employed to capture the global temporal structure. An extra adjusted gate is added on top of temporal attention in the model to guide the decoder in learning a better attention weight. Thus, the model can generate better sentences.
3. Beam search. Other than the greedy search that decides words immediately, Beam Search considers several candidates each time and saves a table of the candidate sequences. After the decoding finishes completely, the sequence that has the highest overall score from a basket of candidate sequences is selected. Theoretically, beam search strategy enlarges the search space, it helps to find a better solution.

We carry out the ablation study to evaluate the model performance with aforementioned component across different metrics. Referring to the result, by learning a video representation with Resnet, ECO and Res3D, adding adjusted gate on top of attention and using beam search (beamsize is 2), the model get better score against the baseline model.

Question 4: For the sake of enhancing the model's sensitiveness to the actions contained in the video, we use res3D to obtain short-term dynamics and ECO to learn the relationship of actions that appear among frames, but both of the models the local temporal structure of the video. To capture the global temporal structure, we employ an attention mechanism. Attention mechanisms in deep neural networks are inspired by human attention that sequentially focuses on the most relevant parts of the information over time. We adopt the temporal attention mechanism [175] to learn a mapping that guides the model to know which set of moments (frames) to look at when generating word sequences. To see if res3D, ECO, and temporal attention make the model more sensitive to action, we conduct the study based on performance across different video categories. Adding eco and c3d features helps the model perform better in many video categories across four tires, as the Msrvtt dataset is rich in action among categories. It has been widely used for action recognition tasks. From this point on, eco and c3d features help the model improve. This implies that they enhance the sensitivity of the model to the actions.

Question 5: From the User Study3, we know that visually impaired people prefer generated sentences by video caption models that are easy and simple to understand. We add an extra adjusted gate to the attention mechanism in the model, which we intuitionally believe can guide the decoder to learn a better attention weight. Thus, the model can generate sentences with better readability. To answer this research question and check the effectiveness of the adjusted gate, we conduct an ablation study. We use several readability metrics to evaluate sentences generated by the models with and without adjusted gates.

According to the study, the video caption model can generate sentence with good readability and it is suitable for our target audience and the public to read easily. By adding an extra adjusted gate to attention mechanism, it increases the model performance in accuracy without adding difficulty in reading.

Question 6: A Latency analysis is conducted to find a proper answer to the Research Question 6: How to reduce the latency of the video caption model? The analysis consists of three parts: 1) the contrast of the latency of the model with different add-on components; 2) the analysis of the relation between video and latency in terms of video length and video's number of frames; 3) the latency contribution of different process and further analysis on outliers. By downsampling and changing the model parameters, the latency of the model has been reduced. At the beginning of this study, the model had a high latency, but then we found out that it was because of a conflict between Pytorch and Windows. An error was caused by setting num_workers, which is the parameter of Dataloader in Pytorch, greater than 0 in the Windows environment. Thus, once we set the num_workers to zero, the model latency meets our requirements.

9

Development and Reflection since 2019

9.1. Introduction

This thesis was started in 2018 and has been interrupted several times over the years due to personal health reasons. This year, I was able to continue working on this thesis again. Since a long time has passed, I have started a new chapter to fill the time gap with a summary of the latest research works. Based on the latest research developments, I will review and reflect on my original research design. At the same time, I propose, based on the current literature review, how I would have designed the research and video captioning model.

This chapter contains two main sections. The first section presents a literature review on video captions, emphasizing the period from late 2023 to the present. The second section illustrates what I will do in this research if I start today.

9.2. Literature Review

In recent years, Artificial Intelligence has experienced a notable transformation, largely driven by the rise of Transformer[154] architectures in Language Models. This architectural innovation, developed by Google in 2017, has significantly influenced Natural Language Processing and then boosted the development of the area of Computer Vision. A prominent early example of this influence is the Vision Transformer (ViT)[183] proposed in 2020, which applies Transformers to partition images into multiple patches, treating each patch as a distinct visual token for input representation.

As a downstream task of computer vision, the video caption domain is also beginning to utilize transformers to improve the model performance. More and more research is going on in this direction. Some literature reviews have covered these works, like [1] and [120], providing a comprehensive survey on video captioning concerning the deep learning approach.

Since 2023, the surge in Large Language Models (LLMs) has naturally led to the development of a new category of generative models called Multimodal Large Language Models (MLLMs). Furthermore, Large Vision Language Models (LVLMs) are a specialized category of Multimodal Models that integrate text and image inputs and generate text outputs. The community of the vision and language has achieved significant progress in developing large image-language models [87][88][86][21][16][15]. While very recently, the video-language community starts to explore and apply Large Language Models for video understanding tasks such as video captioning, etc..

In the following chapters, we will first summarise the development and changes observed in video captioning over the past few years. Next, we will provide a detailed overview of the Transformer model and its significant research contributions to video captioning. The introduction of Transformers has profoundly impacted research in video captioning from 2020 to 2023 and has laid the foundation for developing Large Language Models (LLMs). Subsequently, we will introduce LLMs and their prominent model families. We will then explore the applications of LLMs in the multimodal domain, Multimodal Large Language Models (MLLMs). Finally, we will focus on its specialized category, Visual Language Models (VLLMs). The latest VLLMs on video captioning tasks will be reviewed and discussed. Last, we discussed and analyzed the newest application for the visually impaired.

9.2.1. Video Captioning

In recent years, we have witnessed three main research concern shifts in the video captioning research community.

Firstly inspired by the transformer and further influenced by the breakthrough development of large language models(LLMs), the research community has begun integrating LLMs for visual understanding tasks. Moreover, the exploration of multimodal large language models(MLLMs) and large video language models(LVLMs) is appearing.

The research center has been transferred from the single-sentence video captioning task to the dense video captioning task (which is also known as video description, generating multiple sentences for the target video). This transfer is probably due to the large language models requiring huge data to train them.

The dense video captioning task and the single sentence video captioning task use different datasets to train and metrics to evaluate. Indeed, the change of task arises from the change of datasets and metrics. Further, more and more work is involved in developing better video caption/description benchmarks since the current video captioning benchmarks are relatively simplistic.

Many excellent literature reviews are on video captioning like [1] and [120]. They discuss and summarise the research works in this field before and in 2023. Since the emergence of the LLMs in late 2023, the video-language community has just started to explore the application of LLMs in video captioning. Till now, there has been no survey on this topic. The latest and most relevant reviews are [184], [143], and [11]. The former two reviews concentrated on the general video understanding tasks with LLMs and did not cover video captioning tasks in depth. The third one[11] presents a comprehensive review of the current state of Large Language Models (LLMs) with multimodal capabilities, as well as the latest developments in Multimodal Large Language Models(MLLMs). However, it mainly focused on the historical development of models and did not address the video captioning task, nor did it even address video understanding.

Single Sentence Video Captioning

However, the mainstream video captioning research community has shifted its interest to dense video captioning. There is still some literature on single-sentence video captioning tasks in 2024.[48][103] [174]

In this work[48] introduce a compact encoder-decoder model for video captioning that utilizes CLIP as the encoder and Visual GPT, an adaptation of GPT-2 for text generation, as the decoder. The model operates in two phases: keyframe selection using CLIP and dynamic information encoding. Keyframes are identified by sampling frames that exhibit significant scene transitions, reducing the need for human heuristics. Dynamic information is encoded through the self-attention mechanism of the transformer block. During decoding, Visual GPT enhances the synthesis of visual and textual information. It utilizes cross-attention like GPT-2 but adds a dynamic controller to improve the interaction between caption context and visual data, facilitating better multimodal integration. Notably, CLIP is applied only in the keyframe selection phase. It is kept fixed during training, allowing the model to primarily rely on GPT-2 and a few transformer blocks, thereby reducing resource requirements.

NarrativeBridge[103] is a framework encompassing a benchmark and architecture designed for causal-temporal narrative learning. The Causal-Temporal Narrative (CTN) captions benchmark employs a large language model (LLM) and few-shot prompting to generate video descriptions that clearly express cause-effect relationships and temporal sequences. This benchmark facilitates the generation of cohesive captions demonstrating how events, such as reckless driving leading to a damaged car and subsequent group behavior, are interconnected. By improving models' abilities to convey the significance and order of events, this approach mitigates the limitations of existing datasets. It highlights the need for causal-temporal understanding in video captioning. To ensure caption quality, an automatic evaluation framework is proposed that assesses relevance to the video content, retaining or discarding captions based on a defined scoring threshold.

Similar to [103], in this work [174], the multifaceted video captioning approach is introduced to enhance video-language datasets by improving their modality and context awareness. This approach captures various elements, including entities, actions, speech transcripts, aesthetics, and emotional cues, ensuring a comprehensive representation of video content while allowing flexible alignment between textual descriptions and video representations in multimodal training models. They also explored strategies for using language models to generate high-quality, factual descriptions, creating an agent-like framework to ensure standards are met and to minimize hallucinations, thereby reducing the need for human intervention. Finally, the method's effectiveness on language-video embeddings is evaluated through text-video retrieval tasks. Significant improvements are demonstrated using the MSR-VTT dataset and several multimodal retrieval models without pretraining, attributable to the enhanced dataset. However, this has limitations, including a narrow focus on the MSR-VTT dataset and a limited range of models. Additionally, relying on metrics like recall at k (r@k) may overlook subtle improvements, suggesting a need for human evaluations to provide a more nuanced assessment of dataset quality.

9.2.2. Transformer

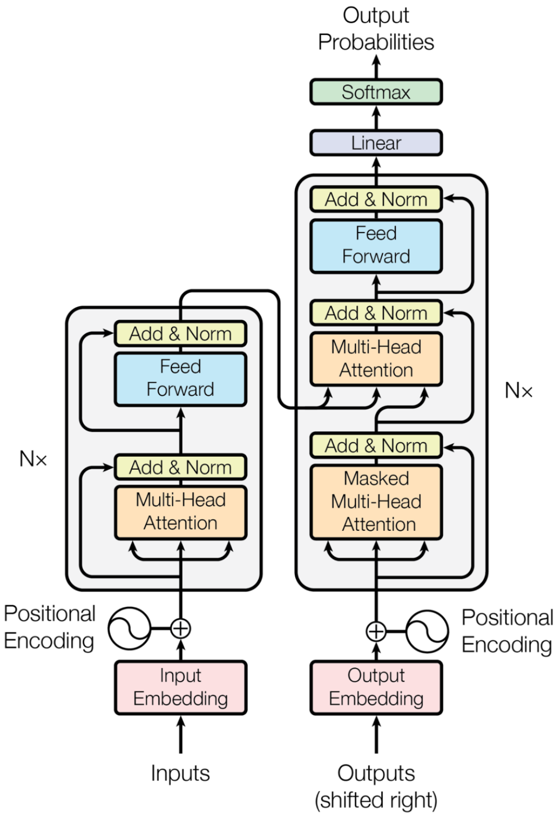


Figure 9.1: The standard transformer inherits the overall encoder-decoder structure from neural sequence models. Both encoder and decoder consist of 6 stacked identical layer blocks. Each block has two sub-layers for the encoder: a multi-head self-attention layer and a position-wise fully connected feed-forward layer. For decoder, each block has the aforementioned two sub-layers in the encoder block and an encoder-decoder attention layer as third sub-layers.[154]

Transformer networks[154] is first designed to tackle machine translation and English constituency parsing tasks in Neural Language Processing (NLP). The standard transformer inherits the overall encoder-decoder structure from neural sequence models, as shown in Fig.9.1. Both encoder and decoder consist of 6 stacked identical layer blocks. Each block has two sub-layers for the encoder: a multi-head self-attention layer and a position-wise fully connected feed-forward layer. For the decoder, each block has two sub-layers above the encoder block, and an encoder-decoder attention layer is a third sub-layer. The novelty of the transformer is that it does not use recurrent connections and only uses an attention mechanism to solve sequence problems. The non-use of recurrent connections gives global information capture and parallel computation an advantage.

Transformer becomes so popular it is not because it avoids the weakness of recurrent networks like being inability to scale up and parallelize, it is because of the propose of BERT [30]. As its name implies, BERT(Bidirectional Encoder Representations from Transformers) is a transformer-based model. It only uses an encoder similar to the standard transformer's encoder, which means it only contains Attention and feed-forward layers. BERT can be trained on the larger dataset unsupervised and then finetuned to the smaller dataset to solve downstream tasks. Since BERT is proposed and its pre-trained BERT model on the sizeable unlabelled text corpus gains excellent performance on a wide range of NLP tasks like question answering and language inference, the transformer-based pretraining method has become very popular across different deep learning research fields. After the propse of BERT, more and more transformer-based models appears.

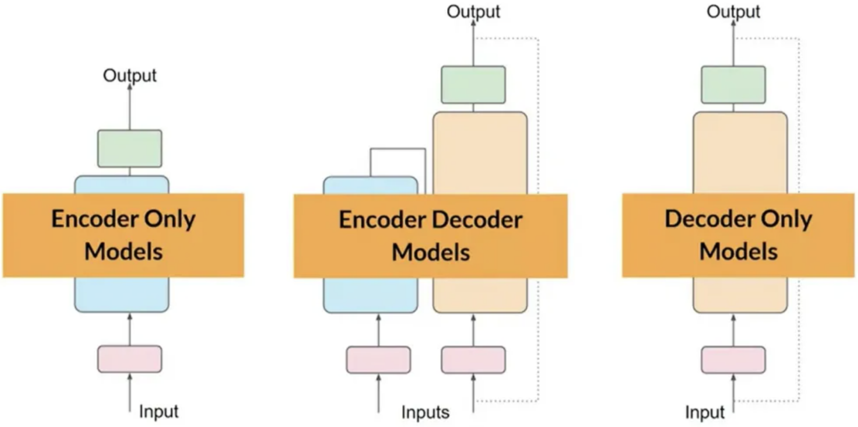


Figure 9.2: the structure of the transformer-based model of three main categories

The structure of the transformer-based models can be classified into three main categories, as shown in Fig. 9.2.

- Encoder-Only (Autoencoding) Models: Encoder-only models, also known as autoencoding models, are pre-trained using a masked language model, in which certain tokens in the input sequence are randomly masked, and the goal of the model is to predict the masked tokens to reconstruct the original sentence. The representative models are: BERT[30], RoBERTa[90]. Since most of MLLM is generative, this structural model is less present in MLLM.
- Decoder-Only (Autoregressive) Models: decoder-only models, pre-trained using a causal language model, aim to predict the next token based on a sequence of previous tokens. This process is also known as complete language modeling. Unlike encoder-only models, decoder-only models mask the input sequence and iteratively predict the next token, creating a unidirectional context. This model type utilizes the original architecture's decoder component without the need for an encoder. Representative models are: GPT[115], BLOOM[168], Qwen[7], LLAMA[149]. This architecture is currently the dominant one in MLLM.
- Encoder-Decoder (Sequence-to-Sequence) Models: Sequence-to-sequence models combine the encoder and decoder components of the original Transformer architecture. Sequence-to-sequence models are useful for translation, summarisation, and question-answer tasks and are represented by T5[119], BART[73].

In addition, the exploration of the Mixture of Experts (MoE) has attracted more and more attention. Compared with the Dense model, sparse architectures can scale up the total parameter size without increasing the computational cost by selectively activating the parameters. That is, the training speed is faster under the same computational resources, and larger models can be trained. Empirically, the MoE implementation performs better than the Dense model on almost all benchmarks. The Representative model is Mixtral 8x7B[54].

The vision community has grown interested in adopting transformer-based methods to solve problems in their area. The research of Dosovitskiy et al.[35] shows that in the image classification task, a Vision Transformer(ViT) performs better than a convolutional neural network. Later in the video classification field, a Video Vision transformer(ViViT) is proposed to encode spatiotemporal tokens extracted from video [6]. Similarly, TimeSformer also uses the transformer architecture to learn spatiotemporal representation with different self-attention schemes[9]. In action recognition, the Video Swin Transformer(VidSwin) attains better results by using an inductive locality bias in the video transformer instead of computing self-attention globally[91].

There are some transformer-based works on video captioning. Lin et al. proposed SwinBERT, an end-to-end transformers for video captioning. As demonstrated in Fig. 9.3, SwinBERT generally consists of two parts: Video Swin Transformer (VidSwin) and Multimodal Transformer Encoder. The pre-trained VidSwin[91] is used to encode densely sampled video frames into video tokens. The Multimodal Transformer Encoder takes video tokens from VidSwin and word tokens from caption description as input and then encodes them to predicted word tokens by masked language modeling. To enhance the efficiency of modeling long video to-

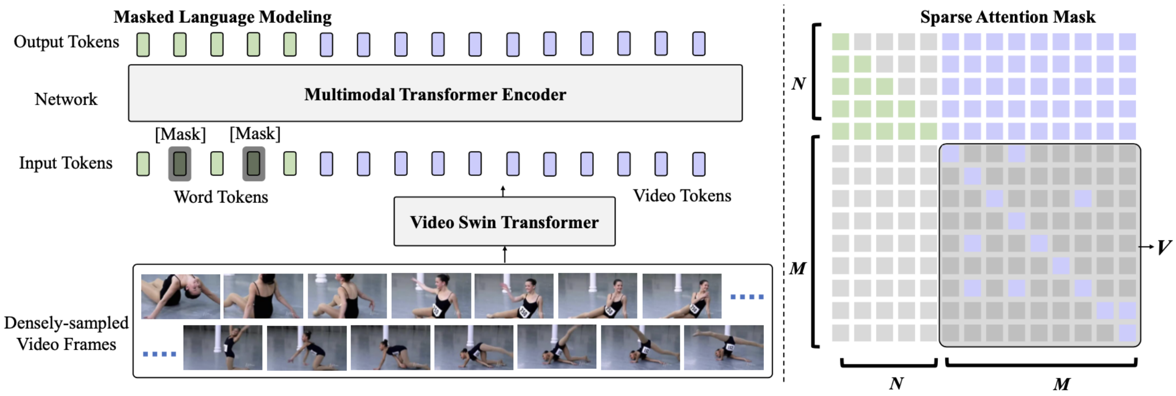


Figure 9.3: In general SwinBERT consists of two parts: Video Swin Transformer (VidSwin), and Multimodal Transformer Encoder. The pre-trained VidSwin[91] encodes densely-sampled video frames into video tokens. The Multimodal Transformer Encoder takes video tokens from VidSwin and word tokens from caption description as input and then encodes them to predicted word tokens by masked language modeling. As shown on the right, a Sparse Attention Mask is employed as the regularizer for the Multimodal Transformer Encoder to enhance the efficiency of modeling long video tokens.[83]

kens, a Sparse Attention Mask is employed as the regularizer for the Multimodal Transformer Encoder. Sparse Attention guides the model in placing more emphasis on tokens that have rich spatial-temporal information. Adding Sparse Attention to SwinBERT is reported to outperform SOTA on many benchmarks.[83]

The pros and cons of SwinBERT are evident. Thanks to the flexibility of the transformer, SwinBERT can process video input at variable lengths. However, it is computational memory intensive since its architecture is considerably complex. It consists of two transformer encoders (VidSwin and BERT), requiring sufficient GPU memory. Besides, in their implementation, the sparse attention mask is not time-efficient because it is realized via an extra learnable embedding.[83]

The transformer-based method has shown superiority in many vision tasks. However, transform-related architecture in video captioning has not been fully explored. There remain some challenges that need further study:

1. Domain gap between pretraining and finetuning. Unlike Convolutional Neural Networks (CNNs), transformers lack inductive biases; thus, they require massive data for pretraining. Accordingly, the quality, diversity, and quantity of the dataset greatly influence the performance of the transformer. In most practice for video captioning tasks, action recognition datasets are used for pretraining. For example, SwinBERT[83] uses the transformer that is pre-trained on Kinetics-600[57] and then is finetuned on video captioning datasets like MSR-VTT[171]. This leads to a domain gap between pretraining and finetuning. As reported in [77], data volume cannot eliminate the domain gap.
2. Efficiency. Transformers face the efficiency problem in terms of quadratic time and memory complexity. Although there are some studies on improving the efficiency of transformers, most of them are from the NLP domain [114][29][68]. Compared to text, video contains much more information. Nevertheless, Video captioning requires a deeper model structure and more extensive parameters to process multimodal information and generate sentences. This adds up to the difficulty of applying transformer-based methods for video captioning in practice.

9.2.3. LLMs

The key module that serves as the brain in MLLM is the large language model (LLM). It is more efficient and practical to use pre-trained models than to train an LLM from scratch. Through large-scale pretraining on web corpora, LLMs have been embedded with rich world knowledge and exhibit strong generalization and inference capabilities.

A large language model (LLM)[100] is a specific category of artificial intelligence (AI) technology capable of recognizing and generating text, among other tasks. "large" refers to the extensive datasets on which these models are trained. LLMs are based on transformer models. Many LLMs leverage vast amounts of text data gathered from the Internet, amounting to thousands or millions of gigabytes.

Self-supervised Pre-training

Pre-training a large language model is conducted through a self-supervised approach that leverages unlabeled data primarily sourced from the internet. This method removes the necessity for manual annotation, allowing the model to autonomously learn the intricate patterns and structures that characterize human language. However, the vast array of online text often contains biases, inaccuracies, and typographical errors, underscoring the importance of implementing a data quality filter. This filter is crucial for eliminating extraneous and harmful data, as such noise can compromise the model's performance instead of enhancing it. Once the filtering process is completed, training can commence. Three distinct variants of the transformer model exist, each designed for specific training objectives: encoder-only, decoder-only, and encoder-decoder. We discussed them in the former section.

Regardless of the transformer variant utilized in the training process, the initial phase remains consistent across all models. The encoder generates embeddings for the tokens, which are vector representations that encapsulate the semantic information associated with each token. This process lays the foundation for subsequent training stages, enabling the model to grasp complex language nuances and contextual relationships effectively.[100]

Supervised Fine-tuning

Fine-tuning is the process of extending the training of a general large model to specialize it for specific tasks. Pre-trained models, while versatile, require fine-tuning to become suitable for particular applications. Early models like BERT[30], trained through self-supervision, could not perform specific tasks without fine-tuning on labeled data, referred to as supervised fine-tuning; for example, BERT was fine-tuned for 11 different tasks. Although recent large language models (LLMs) might function without fine-tuning, they still benefit from task-specific adjustments. The fine-tuning process differs from pre-training, as it employs supervised learning and requires labeled features, thus needing only a few thousand data points to adapt the model, which already has general language knowledge.

A primary goal of fine-tuning is to align model responses with user expectations during instruction-based prompts, a process known as instruction tuning. This involves creating a dataset of prompt-completion pairs, where model predictions are compared to the desired answers (ground truth) to calculate loss, inform gradient calculations, and facilitate weight updates in the network. While effective, instruction fine-tuning can lead to catastrophic forgetting, where the model may lose previously learned knowledge relevant to other tasks while gaining specialization in the new task.[100]

RLHF and RLAIF

Instruction tuning, as previously discussed, brings large language models (LLMs) closer to achieving alignment. However, in many instances, additional steps are necessary to enhance model alignment further and mitigate unintended behaviors. Reinforcement Learning from Human Feedback (RLHF) and Reinforcement Learning from AI Feedback (RLAIF) are widely recognized approaches utilized in optimizing and aligning LLMs. RLHF involves the development of a reward model that learns to align outputs based on feedback provided by human annotators. After tuning, this reward model can evaluate and score multiple outputs according to human preferences for alignment. The insights generated by this reward model are then utilized to refine further and enhance the original large language model (LLM) performance. [100]

Conversely, RLAIF directly connects a pretrained, well-aligned model to the LLM, facilitating its ability to learn from larger, more refined models. This method enhances the LLM's performance by allowing it to benefit from the knowledge and alignment of more advanced systems. RLHF and RLAIF collectively represent significant advancements in natural language processing, as they leverage both human and artificial intelligence feedback to improve model alignment, functionality, and overall effectiveness in various applications.[100]

There are three main families of LLMs: GPT, LlaMA, and PaLM.

The GPT Family

Recent progress in natural language processing (NLP) has paved the way for the creation of robust language models like the GPT[115] and its variants, encompassing large language models (LLM) like ChatGPT. These models undergo pre-training on extensive text datasets and exhibit exceptional performance across various NLP tasks like language translation, text summarization, and question-answering. ChatGPT has showcased its versatility in domains such as education, healthcare, reasoning, text generation, human-machine interaction, and scientific research. A significant milestone in LLM advancement is InstructGPT[107], a framework allowing for instruction finetuning based on Reinforcement Learning from Human Feedback (RLHF). This approach empowers LLMs to adapt to diverse NLP tasks by leveraging human feedback and aligning it with human preferences and values to enhance performance beyond traditional unsupervised pretraining methods.

As a successor to InstructGPT, ChatGPT has integrated these advanced techniques since its launch in December 2022, demonstrating remarkable proficiency in reasoning and text generation. These enhanced NLP capabilities offer extensive applications in fields like education, healthcare, human-machine interaction, and scientific research, driving significant interest and exploration into ChatGPT's potential. Most recently, GPT-4V and GPT-4o have been released. ChatGPT, GPT-4V, and GPT-4o are three distinct variants of OpenAI's advanced language models, each tailored to specific functionalities and applications within the field of artificial intelligence.

GPT or GPT-1 [115], the first model in the Generative Pre-trained Transformer (GPT) series developed by OpenAI, marked a significant advancement in natural language processing (NLP). Released in 2018, GPT-1 demonstrated the effectiveness of the transformer architecture for language modeling. It was trained using unsupervised learning on a diverse dataset comprising millions of web pages, enabling it to learn a broad range of linguistic patterns and generate coherent text based on the input it received.

The architecture of GPT-1 consists of a 12-layer transformer with 117 million parameters. This design allowed the model to capture long-range dependencies in text, a crucial improvement over previous models. GPT-1's training process involved a two-stage approach: pre-training on a large corpus of text data to predict the next word in a sentence and finetuning specific tasks with labeled data. This approach showcased the model's ability to generalize knowledge from pre-training to various downstream tasks.

Despite its relatively modest size compared to its successors, GPT-1 demonstrated the potential of large-scale pre-trained language models. It performed well on several NLP benchmarks, including text classification, question answering, and text generation, setting the stage for developing more advanced models like GPT-2 and GPT-3. GPT-1's success underscored the importance of pre-training and finetuning in creating versatile and powerful language models.

InstructGPT[107] is a specialized variant of OpenAI's language models designed to follow human instructions more accurately and helpfully. Unlike standard GPT models that generate text based on general patterns in the data, InstructGPT is finetuned to understand and adhere closely to user prompts, providing more precise and contextually appropriate responses. This finetuning process involves training the model on a dataset of instructions and desired outputs, allowing it to learn how to respond effectively to specific queries.

The primary goal of InstructGPT is to improve the usability and reliability of AI-generated content, making it more aligned with user intentions. This is particularly useful in applications where precise and accurate responses are critical, such as customer service, education, and content creation. InstructGPT reduces the likelihood of generating irrelevant or off-topic responses by focusing on following instructions, thereby enhancing the overall user experience.

Additionally, InstructGPT incorporates feedback mechanisms where human reviewers rate the model's outputs based on relevance and correctness. This feedback loop further refines the model's ability to generate high-quality, instruction-following responses. Overall, InstructGPT represents a significant advancement in creating AI systems that are not only powerful but also more aligned with human needs and expectations.

ChatGPT[106] is a conversational AI model built on the GPT-3.5 architecture, designed primarily for generating human-like text in interactive settings. Its strength lies in understanding and producing coherent, contextually appropriate responses, making it highly effective for customer support, virtual assistants, educational tools, and general-purpose text generation applications. ChatGPT excels in scenarios where natural language understanding and generation are crucial for user engagement and automated communication.

OpenAI has made substantial advancements in deep learning with the release of GPT-4, a comprehensive multimodal language model designed to process both image and text inputs and generate text outputs. Although it does not yet fully match human capabilities in all real-world contexts, GPT-4 exhibits human-level performance on various professional and academic benchmarks. For example, it scored within the top 10% of test-takers on a simulated bar examination, significantly outperforming GPT-3.5, which scored in the bottom 10%.

The development process of GPT-4 involved six months of iterative alignment, during which insights were incorporated from OpenAI's adversarial testing program and the deployment of ChatGPT. This iterative process has culminated in GPT-4 achieving unprecedented performance regarding factual accuracy, steerability, and compliance with provided guidelines.

GPT-4V[105] expands upon the capabilities of GPT-4 by incorporating multimodal functionalities, enabling the model to process and interpret both text and visual data. This model can generate detailed descriptions of images, answer questions related to visual content, and create images from textual descriptions. GPT-4V is particularly valuable in fields that require a blend of visual and textual understanding, such as image captioning, interactive educational tools, and creative content creation. Its ability to handle both modalities makes it a versatile tool for applications where integrating visual context with language is essential.

GPT-4o[104] is a specialized variant focused on optimization tasks, leveraging the core GPT-4 architecture to solve complex problem-solving scenarios efficiently. Unlike ChatGPT and GPT-4V, GPT-4o is not designed for visual understanding but excels in mathematical and algorithmic applications. It is tailored for industries like logistics, finance, manufacturing, and energy management, where optimization of processes, resource allocation, and decision-making are critical. GPT-4o's capability to generate and refine solutions for optimization problems makes it a powerful tool for enhancing operational efficiency and reducing costs.

Version	Uses	Architecture	Parameter Count	Year
GPT-1	General	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax with Book Corpus: 4.5 GB of text	117 million	2018
GPT-2	General	GPT-1, but with modified normalisation with Web Text: 40 GB of text	1.5 billion	2019
GPT-3	General	GPT-2, but with modification to allow larger scaling with 570 GB plaintext	175 billion	2020
InstructGPT	Conversation	GPT-3 fine-tuned to follow instructions using human feedback model	175 billion	2022
ChatGPT	Dialogue	Uses GPT-3.5, and fine-tuned with both supervised learning and reinforcement learning from human feedback (RLHF)	175 billion	2022
GPT-4	General	Trained with both text prediction and RLHF and accepts both text and images as input, third party data	1.76 trillions	2023
GPT-4V(ision)	General	GPT-4, with an encoder transforming visual modality	1.76 trillions	2023
GPT-4o(mni)	General	GPT-4, processing multi-modal input	1.76 trillions	2024

ChatGPT is optimized for conversational text generation, GPT-4V integrates visual and textual data processing for multimodal applications, and GPT-4o focuses on solving optimization problems across various industries. Each model leverages the foundational GPT-4 architecture to address specific user needs, showcasing the adaptability and wide-ranging potential of OpenAI's AI technologies in diverse domains.

The LLaMA Family

LLaMA[149], developed by Meta AI, is a series of open-source large language models (LLMs) designed to advance research and application development within natural language processing (NLP). The first set of LLaMA models was released in February 2023. Meta AI's objective was to provide the research community with accessible tools to explore the dynamic landscape of LLM applications. The LLaMA series includes foundational models with parameters ranging from 7 billion to 65 billion. Unlike models emphasizing rapid training, LLaMA is optimized for execution on a single GPU, prioritizing inference speed over the reduction of training time.[149]

In July 2023, Meta AI introduced LLaMA-2[150] and LLaMA-2 Chat, updated iterations of the LLaMA model series. These models, with parameter sizes ranging from 7 billion to 70 billion, feature several key improvements. Notably, LLaMA-2 benefits from a 40% larger pretraining corpus and an extended context length increased from 2,048 to 4,096 tokens. A distinct advancement in LLaMA-2 and LLaMA-2 Chat is integrating the finetuning method known as Reinforcement Learning from Human Feedback (RLHF), which differentiates these models from their predecessors.

Meta LLaMA 3[36], the latest iteration in the LLaMA series, was released for widespread use this year. This version includes pretrained and instruction-fine-tuned models with 8 billion and 70 billion parameters, catering to various applications. The LLaMA 3 models mark a substantial advancement over LLaMA 2, establishing new benchmarks for language models at these parameter scales. Advances in pretraining and post-training techniques have led to these models achieving superior performance at the 8B and 70B parameter levels. Specifically, improvements in post-training methodologies have notably decreased false refusal rates, enhanced model alignment, and increased the diversity of responses.

The LLaMA family is expanding rapidly, with an increasing number of instruction-following models being developed based on LLaMA and LLaMA-2, including Alpaca[144] (finetuned from the LLaMA-7B model), Vicuna-13B[23] (by finetuning LLaMA on user-shared conversations), Koala[142] (built on LLaMA and center on long video question answering), just to name a few.

The PaLM Family

Google's PaLM (Pathways Language Model)[26] family is at the forefront of advancements in large language models (LLMs), pushing the limits of language comprehension and generation. While these models are primarily closed-source with limited public availability, they have significantly advanced the capabilities of few-shot learning and complex reasoning tasks.

The original PaLM model was proposed in April 2022 and remained proprietary until March 2023. This large language model (LLM) features 540 billion parameters and is based on a transformer architecture. It underwent pre-training on a comprehensive text corpus containing 780 billion tokens, encompassing a diverse array of natural language tasks and applications.

U-PaLM[146] is a variant of the PaLM model that focuses on improving computational efficiency and is continuously trained to augment its performance and capabilities. By using UL2R[145], a technique designed for the incremental training of large language models (LLMs) utilizing a limited number of steps with UL2's mixture-of-denoiser objective. This training methodology reportedly achieves approximately a 2x computational savings rate.

Furthermore, U-PaLM was enhanced through instruction finetuning, resulting in the development of Flan-PaLM[27]. Unlike the previously mentioned instruction finetuning efforts, Flan-PaLM's finetuning process employs a considerably larger number of tasks, utilizes larger model sizes, and incorporates chain-of-thought data. As a result, Flan-PaLM significantly surpasses the performance of earlier instruction-following models.

PaLM-2[3] is trained using a mixture of objectives. It introduces enhancements in efficiency, multilingual proficiency, reasoning abilities, and performance on downstream tasks compared to its predecessor, PaLM.

Med-PaLM[137] and its successor Med-PaLM 2 are domain-specific PaLMs and are designed to provide high-quality answers to medical questions. Med-PaLM is finetuned on PaLM through instruction prompt tuning, a parameter-efficient technique for aligning large language models (LLMs) to new domains using a limited number of exemplars. While Med-PaLM demonstrates promising results across various healthcare tasks, it remains less effective than human clinicians. Med-PaLM 2 enhances the capabilities of Med-PaLM through domain-specific finetuning and ensemble prompting.

Comparison

We compare these three families from the following perspectives:

- Accessibility. The LLaMA family is open-source, offering broad access to model weights and encouraging collaborative research. Conversely, the GPT and PaLM families are mainly closed-source.
- Specialization and generalization. These model families exhibit a dual trend toward specialization and generalization. Med-PaLM, for instance, highlights how LLMs can revolutionize specialized domains by providing expert-level insights into the medical field. In contrast, models such as GPT-4 show exceptional generalization capabilities, efficiently managing a broad range of tasks without task-specific training.
- Efficiency and scalability. The development of models such as Guanaco based on LLaMA aims to optimize performance while minimizing environmental and economic impacts associated with training large-scale models. Concurrently, the progressive enhancement of models like PaLM-2 reflects an effort to improve computational efficiency, thereby facilitating more sustainable advancements in artificial intelligence.

9.2.4. MLLMs

A Multimodal Large Language Model (MLLM) is defined as a model that integrates the cognitive capabilities of Large Language Models (LLMs), such as GPT-4 and LLaMA-3, with the ability to interpret, reason, and generate responses using multimodal information. This feature enables the model to process and analyze various types of inputs, including both textual and visual data. In May 2024, the launch of OpenAI's GPT-4o attracted considerable media attention. [11]

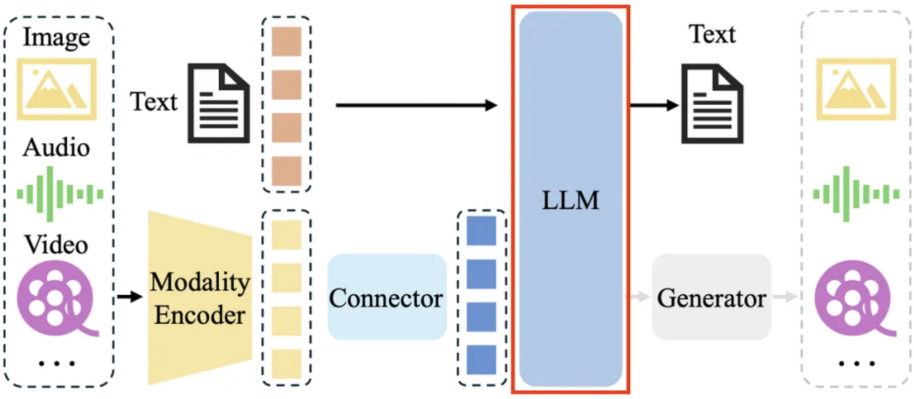


Figure 9.4: the architecture of the classical MLLM

The classical MLLM can be abstracted into three modules, i.e., Modality Encoder, LLM, and Connector, a modal interface that connects them, as shown in fig.9.4. Similar to humans, Modality Encoder such as Image/Audio is the eye/ear that receives and preprocesses the light/acoustic signals, whereas the LLM acts like a human brain that understands and reasons about the processed signals. In between, modal interfaces are used to align the different modalities. Some MLLMs also include a Generator for outputting non-textual modalities.

9.2.5. LVLMs

Large Vision Language Models (LVLMs) are a specialized category of Multimodal Models that integrate text and image inputs and generate text outputs. The community of the vision and language has achieved significant progress in developing large image-language models (Image-LLMs).[87][88][86][21][16][15]

In summary, these models link pre-trained video encoders with pre-trained large language models (LLMs) to achieve zero-shot video understanding capabilities. It normally consists of three primary parts: a visual encoder, a connector, and a large language model (LLM). As its name implies, the visual encoder is used for visual feature extraction. The connector maps the visual and textual information together. The large language model (LLM) is employed to decode the context between these two modalities. LLaVA-1.5[85] is the most representative model, which has significant performance. It employs Vicuna-v1.5[182] as the language decoder and CLIP-Large[117] as the visual encoder. Between them is the connector, which is a two-layer MLP. Inspired by this work, the video-language community started to develop video-LLMs, transforming video into tokens that LLMs can accept to improve the performance of video understanding tasks. [65][81][79] For instance, VideoChat2[65] uses a video transformer and a Q-Former[75] to encode the video content into tokens.

As more video LLMs emerge this year, we will discuss and review what the author feels is the most representative SOTA in the following paragraphs.

As we mentioned in the former section, two literature [103] and [174] are reported to employ LLMs to generate better datasets for the single sentence video captioning. There exists another work [16] that proposes to use LVLMs to generate better video-caption pairs for dense video captioning tasks. Chen et al. developed a ShareGPT4Video[16] dataset, which contains 40,000 high-quality video-caption pairs across various categories. The Differential Sliding-Window Captioning strategy (DiffSW) is proposed as an efficient and scalable approach for generating video captions. It reformulates the all-frames-to-caption task into a differential description task. A detailed caption is first created for the initial frame, followed by applying a sliding window of length two for the subsequent frames. The multimodal model GPT4V identifies changes between frames based on three inputs: the previous frame, its differential caption, and the current frame, capturing variations in camera movement, object movement, character actions, and scene transitions. After generating all differential captions, these are processed by GPT4 to create a cohesive caption for the entire video. It is claimed these captions incorporate extensive world knowledge, object attributes, camera movements, and detailed temporal descriptions of events. Further, Based on the collected dataset, the ShareCaptionor-Video model is obtained upon finetuning the IXC2-4KHD model[34]. In their experiments, ShareGPT4Video enhances the performance of existing large video language models (LVLMs) across various benchmarks, particularly in tasks that necessitate complex temporal understanding. Among the tested LVLMs, VideoLLaVA-7B exhibits the most substantial improvements from the proposed dataset across three comprehensive multimodal video benchmarks. As a result, the final ShareGPT4Video-8B model is constructed upon the LLaVA-Next-8B image multimodal model, consistent with earlier LVLM strategies. However, this model is limited by GPT4V's inability to incorporate audio information concurrently.

Tarsier[160] is a video-LLM with a simple architecture but a carefully designed two-stage training procedure. It has been reported to outperform existing open-source models in video description capabilities significantly. Tarsier model consists of a CLIP-ViT[118] encoder, a projection layer, and a LLM. CLIP-ViT encodes frames individually, and a LLM is employed to capture inter-frame temporal relationships. The training process involves two stages: the first is a multi-task video-to-text pretraining that helps the model comprehend videos from various perspectives. In contrast, the second stage focuses on instruction tuning for generating detailed video descriptions, using high-dynamic videos with well-matched text. Furthermore, a new benchmark for evaluating video description models is designed, which includes a challenging dataset of videos from diverse sources with varying complexities and an automated method for assessing the quality of fine-grained video descriptions. Contrary to Tarsier, Maaz et al. presents comparatively complex video-LLM, a Video Conversation Model called VideoGPT+. Its architectural design comprises four key components:segment-wise sampling, a dual visual encoder(image and video), vision-language adapters, and a large language model. Initially, frames selected through segment-wise sampling are processed by a dual encoder that integrates both image and video encoders, allowing for a comprehensive visual data analysis. The extracted feature sets are then transformed into the language space using vision-language (V-L) adapters, which facilitate the alignment of visual and textual representations and project visual features into the language domain. These tokens undergo adaptive token pooling to manage dimensionality and enhance representational efficiency before being concatenated and input into the LLM. Although the model's architecture is considerably complicated, the introduction of segment-wise sampling reduces the complexity of computation. Video encoders often encounter computational limitations, restricting their processing capabilities to sparse frames. Uniform sampling exacerbates the computational complexity of self-attention, as it necessitates attending to features from all frames. Furthermore, video encoders are generally trained on sparse frames, and increasing the number of processed frames can impair their ability to capture temporal information accurately. In contrast, the segment-wise sampling approach partitions the video into smaller, more manageable segments, allowing the encoder to extract rich temporal cues within each segment efficiently.

Live video captioning is an emerging and complex issue that has not been extensively explored in the scientific literature. Current SOTA models generally operate on a fixed number of down-sampled frames, producing a single prediction only after processing the entire video. In response, [185] is proposed as a streaming dense video captioning model featuring two novel components: a clustering-based memory module is introduced that can manage arbitrarily long videos while maintaining a fixed memory size; a streaming decoding algorithm is developed that allows the model to generate predictions before completing the analysis of the entire video. Streaming video captioning[185] is limited to supporting captions and does not allow for free-form dialogue. Additionally, its temporal regions for captioning are fixed, which reduces its flexibility and generality compared to VideoLLM [14]. Chen et al. introduce Learning-In-Video-strEam (LIVE), a framework that enables large language models (LLMs) to process streaming video effectively. This framework facilitates the generation of temporally aligned responses, accommodates extended video duration, and ensures high inference efficiency. The online VideoLLM model developed within the LIVE framework comprises three core components: the CLIP ViT-L encoder as the image encoder to extract video frame embeddings at a rate of 2 frames per second; an MLP projector to convert embeddings into frame tokens and interleave with language tokens to create input for the LLM; a large language model, specifically Llama-2 or Llama-3.

Very recently, VideoLLaMA 2[22] builds upon its predecessor by incorporating a custom Spatial-Temporal Convolution (STC) connection in place of Q-Former, effectively capturing the spatial and temporal dynamics of video data while producing fewer video tokens. The model employs image-level CLIP (ViT-L/14) as its vision backbone, facilitating compatibility with various frame sampling strategies and enabling flexible aggregation of video features. Additionally, an Audio Branch is integrated through joint training, enhancing the model's multimodal understanding by incorporating audio cues. For video captioning, experiments are conducted using the Multi-Source Video Caption (MSVC) benchmark, which comprises 500 videos with human-annotated captions from MSVD, MSRVTT, and VATEX to ensure diverse scenarios. A ChatGPT-assisted evaluation is employed to assess both generated and human-annotated captions for accuracy and detail. Results show that VideoLLaMA 2 demonstrates competitive performance among open-source models and approaches the quality of some proprietary models across multiple benchmarks. As far as this review has been done, VideoLLaMA is the only work that incorporates video and audio for video captioning tasks.

Evaluating LLMs and LVLMs is challenging yet essential for their advancement and enhancement. Over the past year, benchmark scores for various tasks have consistently improved; however, the practical performance of these models remains unsatisfactory. This issue stems from inadequate evaluation methods that do not accurately assess the applicability of the models in real-world scenarios. The latest work [15] has disclosed two main issues present in current benchmarks. First, visual content is often unnecessary for many samples, as answers can frequently be inferred from the questions and options or derived from the general knowledge embedded in large language models (LLMs). This is a common characteristic observed across existing benchmarks. Second, unintentional data leakage occurs during LLMs and large video language models (LVLMs) training. These models can answer visual-dependent questions without the corresponding visual inputs, indicating that they may have memorized these examples from their extensive training datasets. With the aforementioned two observations, Chen et al. proposes MMStar, a new benchmark for evaluating LVLM.

9.2.6. Application

Be My Eyes is a mobile application designed to support visually impaired and blind individuals by connecting them with sighted volunteers for visual assistance through a live video call. Since its launch in 2015, the application has grown exponentially, fostering a global community that bridges the gap between sighted helpers and those in need of visual aids.

The core concept of Be My Eyes is straightforward. When visually impaired users need help with tasks requiring eyesight, such as reading labels, navigating unfamiliar environments, or troubleshooting technical issues, they can initiate a video call via the app. The call is routed to a sighted volunteer who can see through the user's camera and provide real-time assistance.

Be My Eyes now has integrated with GPT-4 technology. The application added a new function called Be My AI. The GPT-4 is an automated assistance in scenarios where a human volunteer is not immediately available. For instance, when a visually impaired user needs help identifying objects or reading text, Be My AI, which is actually GPT-4, can analyze images taken by the user and provide descriptions or transcriptions in real time. Besides, the GPT-4 enhances the experience of interactions. GPT-4 can handle routine queries and offer detailed explanations.

A small group of employees tested the beta version of the GPT-backed assistant in early February 2024. In summary, Be My Eyes leverages the power of community and AI to provide double support to visually impaired users. While the current release still has some limitations:

1. First of all, GPT-4 is not designed especially for people with low vision. Secondly, Be My AI has not been tested on people with low vision. According to our findings, people who are born blind and those who turn blind in their older age perceive the world very differently. People with acquired blindness perceive the world similarly to sighted people. On the contrary, people who are born blind perceive the world differently. For example, color is not very important to them since they do not have the concept of color.
2. Be My AI processes a single image at once other than video streaming.
3. Be My AI processes images and provides very detailed descriptions by default. Thus, it takes a high latency. As I have witnessed, the latency is usually over 5 seconds.

9.2.7. Summary

While image-large language models (image-LLMs) have achieved impressive capabilities, video-large language models (video-LLMs) have not progressed similarly due to inherent complexities. Unlike static images, videos feature temporal dynamics and synchronous audio streams, significantly enhancing their information content. This combination complicates extracting and interpreting meaningful patterns, increasing data complexity and creating unique computational challenges.

Although the introduction of LLMs benefits video understanding and its downstream tasks like video captioning, the primary difficulty in video understanding still lies in managing temporal dynamics—recognizing visual patterns, adapting to changes over time, and correlating these with audio inputs. These challenges hinder accurate future predictions and comprehension of complex scenarios, such as interactions among multiple entities.

Moreover, current Video-LLMs are limited in their ability to process temporal dynamics effectively, failing to leverage the available information across frames, which impedes accurate event predictions. They often overlook audio integration, a crucial source of contextual cues for comprehensive scene understanding. These shortcomings underscore the need for more advanced video LLMs capable of addressing the complexities of multimodal video data while preserving processing efficiency and contextual integrity.

9.3. What I will do if I start the thesis nowadays

Based on the literature review and the target audience's needs, I will propose a new method and cite the related work below.

- Live streaming framework. we will adopt the Learning-In-Video-Stream (LIVE) framework [14] for the video streaming and captioning. This novel approach facilitates temporally aligned long-context and real-time dialogue within continuous video streams. It encompasses a variety of comprehensive strategies to enable dialogue in video streaming, including a training objective tailored for language modeling with continuous streaming inputs and an optimized inference pipeline aimed at enhancing model response times in real-world video environments.

- Live streaming model: A live streaming model is built upon the live streaming framework. It consists of three components: an image encoder, a connector, and a LLM as a decoder.
  1) Image encoder: CLIP-ViT-L/14[118] encoder (pre-trained) is used for image encoding. The base model CLIP incorporates a ViT-L/14 Transformer architecture as the image encoder alongside a masked self-attention Transformer functioning as the text encoder. These encoders are trained to enhance the similarity between (image, text) pairs by applying a contrastive loss function.
  2) We utilize the Spatial-Temporal Convolution (STC) connection proposed in [22] as a Connector. Initially, video frames are encoded into feature representations on a frame-by-frame basis. These encoded features are then processed through the STC connector, comprising two spatial interaction modules and one spatial-temporal aggregation module. The STC connector ensures that the spatial-temporal order of the output visual tokens is preserved, which is essential because large language models (LLMs) depend on consistent token order during training and inference. Furthermore, the RegStage block is employed prior to and following spatial-temporal down-sampling to reduce information loss throughout the process.
  3) The recently released Llama 3.1-Minitron-4B[102], a pruned and distilled version of Llama 3.1 [36] developed by researchers at NVIDIA, is employed as the decoder. By using structured weight pruning alongside knowledge distillation, Minitron achieves a 1.8x reduction in compute costs for training the full model family (15B, 8B, and 4B) while performing comparably to other community models, such as Llama-3 8B, and surpassing the state-of-the-art compression techniques documented in the literature.

- Offline model: Current SOTA models generally operate on a fixed number of down-sampled frames, producing a single prediction only after processing the entire video. In order to compare the performance, an offline model is built with a video encoder in addition to the live streaming model. InternVideo2[164] is employed as the video encoder for extracting the temporal information.

- Audio: Audio is still not considered in our work. While including audio information may intuitively enhance video understanding, our target users are more likely to utilize this model in noisy environments to obtain contextual information. Background noise could interfere with the model's video comprehension in such settings. Furthermore, integrating an audio module would increase the overall model size and computational resource requirements, leading to increased latency.

- Dataset: Many literature[174] [103][16] reported that better-annotated datasets lead to better performance on the video captioning task. We will utilize the recently introduced MultiSource Video Caption (MSVC) benchmark as presented in [22]. This dataset comprises 500 videos with human-annotated captions drawn from MSVD, MSR-VTT, and VATEX, ensuring a diverse range of scenarios and domains.

- Evaluation: Traditional evaluation metrics predominantly focus on exact match statistics between generated and ground truth captions, which often fail to encapsulate the complexity and richness of video content fully. To address this limitation, we integrate a ChatGPT-assisted evaluation approach, similar to the methodology outlined in [94]. This complementary evaluation strategy will allow for a more nuanced assessment of generated captions, considering contextual relevance, coherence, and the overall quality of the descriptions. By combining traditional metrics(including readability metrics) and large language model evaluations, we aim to understand captioning performance in the context of video content comprehensively.

- Latency: There are some techniques that can be used in the model to reduce the latency. Maintaining a continuous key-value cache as the input progresses allows the inference to be sped up Since the decoding process of language models is time-consuming. Parallelizing the fast video frame encoder and the slower language model can avoid the bottleneck in the latter.

Conclusion

Due to significant advancements in the field of artificial intelligence over the past few years, particularly with the emergence of transformers and large language models(LLMs), there has been a marked development in the area of video understanding. Suppose I start my thesis today and design a video captioning model for the visually impaired. In that case, I will use the pre-trained transformer-based visual encoder to extract video features and employ powerful pre-trained LLM to generate captions. Using the recently proposed live-streaming video dialogue framework as a reference, we can design a live-streaming video captioning model. Further, an offline model will also be built to compete with the SOTA model since most existing models only provide captions after processing the entire video. An extra video encoder is added to the offline model to help the understanding of the video dynamics.

Tailor to the requirements of the visually impaired, the MultiSource Video Caption dataset is selected, which is collected based on the MSR-VTT dataset(we previously used) and two other datasets. In addition to the evaluation metrics we employed before(captioning and readability metrics), a ChatGPT-assisted evaluation will be introduced to understand captioning performance comprehensively.

As we expect to build a real-time captioning model for people with vision challenges, the latency is still a vital problem. Several possible ways can be used to avoid high latency: 1) choosing pruned and distilled LLM for caption generation, 2) maintaining a key-value cache to reduce inference time, mitigating potential bottlenecks in the processing pipeline by parallelizing visual encoder and the language decoder.

Bibliography

[1] Moloud Abdar, Meenakshi Kollati, Swaraja Kuraparthi, Farhad Pourpanah, Daniel McDuff, Mohammad Ghavamzadeh, Shuicheng Yan, Abduallah Mohamed, Abbas Khosravi, Erik Cambria, and Fatih Porikli. A review of deep learning for video captioning, 2023. URL `https://arxiv.org/abs/2304.11431`.

[2] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: semantic propositional image caption evaluation. *CoRR*, abs/1607.08822, 2016. URL `http://arxiv.org/abs/1607.08822`.

[3] Rohan Anil, Andrew M. Dai, and et al. Palm 2 technical report, 2023. URL `https://arxiv.org/abs/2305.10403`.

[4] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL `https://openreview.net/forum?id=Hk4_qw5xe`.

[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/arjovsky17a.html`.

[6] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. *CoRR*, abs/2103.15691, 2021. URL `https://arxiv.org/abs/2103.15691`.

[7] Jinze Bai, Shuai Bai, and et al. Qwen technical report, 2023. URL `https://arxiv.org/abs/2309.16609`.

[8] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*, 2005.

[9] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *CoRR*, abs/2102.05095, 2021. URL `https://arxiv.org/abs/2102.05095`.

[10] Alexy Bhowmick and Shyamanta Hazarika. An insight into assistive technology for the visually impaired and blind people: state-of-the-art and future trends. *Journal on Multimodal User Interfaces*, 11:1–24, 01 2017. doi: 10.1007/s12193-016-0235-6.

[11] Kilian Carolan, Laura Fennelly, and Alan F. Smeaton. A review of multi-modal large language and vision models, 2024. URL `https://arxiv.org/abs/2404.01322`.

[12] Chen Chen, Shuai Mu, Wanpeng Xiao, Zexiong Ye, Liesi Wu, Fuming Ma, and Qi Ju. Improving image captioning with conditional generative adversarial nets. *CoRR*, abs/1805.07112, 2018. URL `http://arxiv.org/abs/1805.07112`.

[13] David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 190–200, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9. URL `http://dl.acm.org/citation.cfm?id=2002472.2002497`.

[14] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video, 2024. URL `https://arxiv.org/abs/2406.11816`.

[15] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models?, 2024. URL `https://arxiv.org/abs/2403.20330`.

[16] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, Li Yuan, Yu Qiao, Dahua Lin, Feng Zhao, and Jiaqi Wang. Sharegpt4video: Improving video understanding and generation with better captions, 2024. URL `https://arxiv.org/abs/2406.04325`.

[17] Shaoxiang Chen and Yu-Gang Jiang. Motion guided spatial attention for video captioning. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[18] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016. URL `http://arxiv.org/abs/1606.03657`.

[19] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015. URL `http://arxiv.org/abs/1504.00325`.

[20] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. *CoRR*, abs/1803.01457, 2018. URL `http://arxiv.org/abs/1803.01457`.

[21] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks, 2024. URL `https://arxiv.org/abs/2312.14238`.

[22] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms, 2024. URL `https://arxiv.org/abs/2406.07476`.

[23] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

[24] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL `https://www.aclweb.org/anthology/D14-1179`.

[25] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014. URL `http://arxiv.org/abs/1406.1078`.

[26] Aakanksha Chowdhery, Sharan Narang, and Jet al. Palm: Scaling language modeling with pathways, 2022. URL `https://arxiv.org/abs/2204.02311`.

[27] Hyung Won Chung, Le Hou, Shayne Longpre, and et al. Scaling instruction-finetuned language models, 2022. URL `https://arxiv.org/abs/2210.11416`.

[28] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[29] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. *CoRR*, abs/1807.03819, 2018. URL `http://arxiv.org/abs/1807.03819`.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL `https://arxiv.org/abs/1810.04805`.

[31] Aniqa Dilawari, Muhammad Usman Ghani Khan, Yasser D. Al-Otaibi, Zahoor-ur Rehman, Atta-ur Rahman, and Yunyoung Nam. Natural language description of videos for smart surveillance. *Applied Sciences*, 11(9), 2021. ISSN 2076-3417. doi: 10.3390/app11093730. URL https://www.mdpi.com/2076-3417/11/9/3730.

[32] Pierre L. Dognin, Igor Melnyk, Youssef Mroueh, Jarret Ross, and Tom Sercu. Improved image captioning with adversarial semantic alignment. *CoRR*, abs/1805.00063, 2018. URL http://arxiv.org/abs/1805.00063.

[33] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389, 2014. URL http://arxiv.org/abs/1411.4389.

[34] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, Hang Yan, Yang Gao, Zhe Chen, Xinyue Zhang, Wei Li, Jingwen Li, Wenhai Wang, Kai Chen, Conghui He, Xingcheng Zhang, Jifeng Dai, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd, 2024. URL https://arxiv.org/abs/2404.06512.

[35] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. URL https://arxiv.org/abs/2010.11929.

[36] Abhimanyu Dubey, Abhinav Jauhri, and et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[37] Desmond Elliott and Frank Keller. Comparing automatic evaluation measures for image description. *ACL*, 2014.

[38] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[39] Jenkins J. J. Farr, J. N. and D. G. (Paterson. Simplification of flesch reading ease formula. *Journal of Applied Psychology*, 35:333–337, 1951.

[40] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969125.

[41] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. pages 2712–2719, 12 2013. doi: 10.1109/ICCV.2013.337.

[42] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017. URL http://arxiv.org/abs/1704.00028.

[43] Robert Gunning. The fog index after twenty years. *Journal of Business Communication*, 6:3–13, 1969.

[44] Tszhang Guo, Shiyu Chang, Mo Yu, and Kun Bai. Improving reinforcement learning based image captioning with natural language prior. *CoRR*, abs/1809.06227, 2018. URL http://arxiv.org/abs/1809.06227.

[45] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? *CoRR*, abs/1711.09577, 2017. URL http://arxiv.org/abs/1711.09577.

[46] Hartato, Riandy Juan Albert Yoshua, Husein, Agelius Garetta, and Harco Leslie Hendric Spits Warnars. Technology for disabled with smartphone apps for blind people. In I. Jeena Jacob, Selvanayaki Kolandapalayam Shanmugam, and Robert Bestak, editors, *Expert Clouds and Applications*, pages 271–282, Singapore, 2022. Springer Nature Singapore. ISBN 978-981-19-2500-9.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

[48] Yoonseok Heo, Taehoon Kim, Seunghwan Kim, Jungyun Seo, and Juae Kim. Towards human-interactive controllable video captioning with efficient modeling. *Mathematics*, 12(13), 2024. ISSN 2227-7390. doi: 10.3390/math12132037. URL `https://www.mdpi.com/2227-7390/12/13/2037`.

[49] Marion Hersh and Michael Johnson. On modelling assistive technology systems - part i: Modelling framework. *Technology and Disability*, 20, 10 2008. doi: 10.3233/TAD-2008-20303.

[50] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[51] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Kazuhiro Sumi, John R. Hershey, and Tim K. Marks. Attention-based multimodal fusion for video description. *CoRR*, abs/1701.03126, 2017. URL `http://arxiv.org/abs/1701.03126`.

[52] Mengqiu Hu, Yang Yang, Fumin Shen, Ning Xie, and Heng Tao Shen. Hashing with angular reconstructive embeddings. *IEEE Transactions on Image Processing*, 27:545–555, 2018.

[53] Saiful Islam, Aurpan Dash, Ashek Seum, Amir Raj, Tonmoy Hossain, and Faisal Shah. Exploring video captioning techniques: A comprehensive survey on deep learning method. 2021. doi: 10.1007/s42979-021-00487-x. URL `https://doi.org/10.1007/s42979-021-00487-x`.

[54] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL `https://arxiv.org/abs/2401.04088`.

[55] E Dale JS Chall. *The new Dale-Chall readability formula.* Brookline Books, Cambrige, 1995.

[56] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[57] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Apostol Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[58] Akif Khan and Shah Khusro. An insight into smartphone-based assistive solutions for visually impaired and blind people – issues, challenges and opportunities. *Universal Access in the Information Society*, 19:1–25, 06 2021. doi: 10.1007/s10209-020-00733-8.

[59] Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. Re-evaluating automatic metrics for image captioning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 199–209. Association for Computational Linguistics, 2017. URL `http://aclweb.org/anthology/E17-1019`.

[60] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[61] Atsuhiro Kojima, Takeshi Tamura, and Kunio 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, Nov 2002. ISSN 1573-1405. doi: 10.1023/A:1020346032608. URL `https://doi.org/10.1023/A:1020346032608`.

[62] Ranjay Krishna, Kenji Hata, Frederic Ren, Fei-Fei Li, and Juan Carlos Niebles. Dense-captioning events in videos. *CoRR*, abs/1705.00754, 2017. URL `http://arxiv.org/abs/1705.00754`.

[63] Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond Mooney, Kate Saenko, and Sergio Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of the Workshop on Vision and Natural Language Processing*, pages 10–19, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W13-1302`.

[64] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso A. Poggio, and Thomas Serre. Hmdb: A large video database for human motion recognition. *2011 International Conference on Computer Vision*, pages 2556–2563, 2011.

[65] Yi Wang Yizhuo Li Wenhai Wang Ping Luo Yali Wang Limin Wang KunChang Li, Yinan He and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[66] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From word embeddings to document distances. In *ICML*, 2015.

[67] Hanock Kwak and Byoung-Tak Zhang. Generating images part by part with composite generative adversarial networks. *CoRR*, abs/1607.05387, 2016. URL `http://arxiv.org/abs/1607.05387`.

[68] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019. URL `http://arxiv.org/abs/1909.11942`.

[69] G. Harry Mc Laughlin. Smog grading–a new readability formula. *Journal of Reading*, 12:639–646, 1969.

[70] DiMuzio J. Beauchamp B. et al. LeBrun, M. Evaluating the health literacy burden of canada's public advisories: A comparative effectiveness study on clarity and readability. *Drug Saf*, 36:1179–1187, 2013. doi: https://doi.org/10.1007/s40264-013-0117-8.

[71] Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. ViLBERTScore: Evaluating image caption using vision-and-language BERT. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 34–39, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.eval4nlp-1.4. URL `https://aclanthology.org/2020.eval4nlp-1.4`.

[72] Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Trung Bui, and Kyomin Jung. UMIC: an unreferenced metric for image captioning via contrastive learning. *CoRR*, abs/2106.14019, 2021. URL `https://arxiv.org/abs/2106.14019`.

[73] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. URL `https://arxiv.org/abs/1910.13461`.

[74] Dianqi Li, Xiaodong He, Qiuyuan Huang, Ming-Ting Sun, and Lei Zhang. Generating diverse and accurate visual captions by comparative adversarial learning. 04 2018.

[75] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[76] Lijun Li and Boqing Gong. End-to-end video captioning with multitask reinforcement learning. *CoRR*, abs/1803.07950, 2018. URL `http://arxiv.org/abs/1803.07950`.

[77] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: hierarchical encoder for video+language omni-representation pre-training. *CoRR*, abs/2005.00200, 2020. URL `https://arxiv.org/abs/2005.00200`.

[78] Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. Pytorch distributed: Experiences on accelerating data parallel training. *CoRR*, abs/2006.15704, 2020. URL `https://arxiv.org/abs/2006.15704`.

[79] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023.

[80] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *arXiv preprint arXiv:2203.02053*, 2022. URL `https://arxiv.org/abs/2203.02053`.

[81] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[82] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004. URL `http://aclweb.org/anthology/W04-1013`.

[83] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. *CoRR*, abs/2111.13196, 2022. URL `https://arxiv.org/abs/2111.13196`.

[84] Fenglin Liu, Xuancheng Ren, Yuanxin Liu, Kai Lei, and Xu Sun. Exploring and distilling cross-modal information for image captioning. *CoRR*, abs/2002.12585, 2020. URL `https://arxiv.org/abs/2002.12585`.

[85] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[86] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL `https://arxiv.org/abs/2304.08485`.

[87] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024. URL `https://arxiv.org/abs/2310.03744`.

[88] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

[89] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Optimization of image description metrics using policy gradient methods. *CoRR*, abs/1612.00370, 2016. URL `http://arxiv.org/abs/1612.00370`.

[90] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL `https://arxiv.org/abs/1907.11692`.

[91] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *CoRR*, abs/2106.13230, 2021. URL `https://arxiv.org/abs/2106.13230`.

[92] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0164-8. URL `http://dl.acm.org/citation.cfm?id=850924.851523`.

[93] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via A visual sentinel for image captioning. *CoRR*, abs/1612.01887, 2016. URL `http://arxiv.org/abs/1612.01887`.

[94] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models, 2024. URL `https://arxiv.org/abs/2306.05424`.

[95] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arxiv*, 2024. URL `https://arxiv.org/abs/2406.09418`.

[96] Dewarthi Mahajan, Sakshi Bhosale, Yash Nighot, and Madhuri Tayal. Review of video captioning methods. *International Journal of Next-Generation Computing*, 11 2021. doi: 10.47164/ijngc.v12i5.458.

[97] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2009.

[98] Heidi Anne E. Mesmer. *Tools for Matching Readers to Texts: Research-Based Practices.* The Guilford Press, 2007. ISBN 1593855974.

[99] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA, 2013. Curran Associates Inc. URL `http://dl.acm.org/citation.cfm?id=2999792.2999959`.

[100] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2024. URL `https://arxiv.org/abs/2402.06196`.

[101] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL `http://arxiv.org/abs/1411.1784`.

[102] Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. Compact language models via pruning and knowledge distillation, 2024. URL `https://arxiv.org/abs/2407.14679`.

[103] Asmar Nadeem, Faegheh Sardari, Robert Dawes, Syed Sameed Husain, Adrian Hilton, and Armin Mustafa. Narrativebridge: Enhancing video captioning with causal-temporal narrative, 2024. URL `https://arxiv.org/abs/2406.06499`.

[104] OpenAI. Gpt-4o. . URL `https://openai.com/index/hello-gpt-4o/`.

[105] OpenAI. Gpt-4v(ision) system card. . URL `https://cdn.openai.com/papers/GPTV_System_Card.pdf`.

[106] OpenAI. Chatgpt. 2023. URL `https://chat.openai.com`.

[107] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL `https://arxiv.org/abs/2203.02155`.

[108] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. *CoRR*, abs/1505.01861, 2015. URL `http://arxiv.org/abs/1505.01861`.

[109] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL `https://doi.org/10.3115/1073083.1073135`.

[110] Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach. Adversarial inference for multi-sentence video description. *CoRR*, abs/1812.05634, 2018. URL `http://arxiv.org/abs/1812.05634`.

[111] Ramakanth Pasunuru and Mohit Bansal. Reinforced video captioning with entailment rewards. *CoRR*, abs/1708.02300, 2017. URL `http://arxiv.org/abs/1708.02300`.

[112] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. *CoRR*, abs/1905.03966, 2019. URL `http://arxiv.org/abs/1905.03966`.

[113] Sang Phan, Gustav Eje Henter, Yusuke Miyao, and Shin'ichi Satoh. Consensus-based sequence training for video captioning. *CoRR*, abs/1712.09532, 2017. URL `http://arxiv.org/abs/1712.09532`.

[114] Sai Prasanna, Anna Rogers, and Anna Rumshisky. When BERT plays the lottery, all tickets are winning. *CoRR*, abs/2005.00561, 2020. URL `https://arxiv.org/abs/2005.00561`.

[115] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. URL https://api.semanticscholar.org/CorpusID:49313245.

[116] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016.

[117] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[118] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[119] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

[120] Ghazala Rafiq, Muhammad Rafiq, and Gyu Sang Choi. Video description: A comprehensive survey of deep learning approaches. *Artificial Intelligence Review*, 56:1–80, 04 2023. doi: 10.1007/s10462-023-10414-6.

[121] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732, 2016.

[122] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *CoRR*, abs/1605.05396, 2016. URL http://arxiv.org/abs/1605.05396.

[123] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics (TACL)*, 1:25–36, 2013.

[124] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *CoRR*, abs/1612.00563, 2016. URL http://arxiv.org/abs/1612.00563.

[125] Engelhardt P.E. Rivero-Contreras, M. and D. Saldaña. An experimental eye-tracking study of text adaptation for readers with dyslexia: effects of visual support and word frequency. *Ann. of Dyslexia*, 71: 170–187, 2021. doi: https://doi.org/10.1007/s11881-021-00217-1.

[126] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail. In *German Conference on Pattern Recognition (GCPR)*, September 2014. Oral.

[127] Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. The long-short story of movie description. *CoRR*, abs/1506.01698, 2015. URL http://arxiv.org/abs/1506.01698.

[128] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[129] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. *CoRR*, abs/1809.02156, 2018. URL http://arxiv.org/abs/1809.02156.

[130] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. Translating video content to natural language descriptions. *2013 IEEE International Conference on Computer Vision*, pages 433–440, 2013.

[131] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, pages 1–28, 2015. ISSN 0920-5691. doi: 10.1007/s11263-015-0851-8. URL `http://dx.doi.org/10.1007/s11263-015-0851-8`.

[132] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, Nov 2000. ISSN 1573-1405. doi: 10.1023/A:1026543900054. URL `https://doi.org/10.1023/A:1026543900054`.

[133] Hobin Ryu, Sunghun Kang, Haeyong Kang, and Chang D. Yoo. Semantic grouping network for video captioning. *CoRR*, abs/2102.00831, 2021. URL `https://arxiv.org/abs/2102.00831`.

[134] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, ICPR '04, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2128-2. doi: 10.1109/ICPR.2004.747. URL `http://dx.doi.org/10.1109/ICPR.2004.747`.

[135] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, jan 2016. ISSN 0028-0836. doi: 10.1038/nature16961.

[136] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–, October 2017. URL `http://dx.doi.org/10.1038/nature24270`.

[137] Karan Singhal, Tao Tu, and et al. Towards expert-level medical question answering with large language models, 2023. URL `https://arxiv.org/abs/2305.09617`.

[138] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[139] Liang Sun, Bing Li, Chunfeng Yuan, Zhengjun Zha, and Weiming Hu. Multimodal semantic attention network for video captioning. *CoRR*, abs/1905.02963, 2019. URL `http://arxiv.org/abs/1905.02963`.

[140] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction.* Adaptive computation and machine learning. MIT Press, 1998. ISBN 0262193981. URL `http://www.worldcat.org/oclc/37293240`.

[141] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. URL `http://arxiv.org/abs/1409.4842`.

[142] Reuben Tan, Ximeng Sun, Ping Hu, Jui hsien Wang, Hanieh Deilamsalehy, Bryan A. Plummer, Bryan Russell, and Kate Saenko. Koala: Key frame-conditioned long video-llm, 2024. URL `https://arxiv.org/abs/2404.04346`.

[143] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, Ali Vosoughi, Chao Huang, Zeliang Zhang, Feng Zheng, Jianguo Zhang, Ping Luo, Jiebo Luo, and Chenliang Xu. Video understanding with large language models: A survey, 2024. URL `https://arxiv.org/abs/2312.17432`.

[144] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[145] Yi Tay, Jason Wei, Hyung Won Chung, Vinh Q. Tran, David R. So, Siamak Shakeri, Xavier Garcia, Huaixiu Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, Denny Zhou, Donald Metzler, Slav Petrov, Neil Houlsby, Quoc V. Le, and Mostafa Dehghani. Transcending scaling laws with 0.1 URL https://arxiv.org/abs/2210.11399.

[146] Yi Tay, Jason Wei, Hyung Won Chung, Vinh Q. Tran, David R. So, Siamak Shakeri, Xavier Garcia, Huaixiu Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, Denny Zhou, Donald Metzler, Slav Petrov, Neil Houlsby, Quoc V. Le, and Mostafa Dehghani. Transcending scaling laws with 0.1 URL https://arxiv.org/abs/2210.11399.

[147] Jesse Thomason, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Raymond Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1218–1227, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C14-1115.

[148] Atousa Torabi, Christopher J. Pal, Hugo Larochelle, and Aaron C. Courville. Using descriptive video services to create a large data source for video annotation research. *CoRR*, abs/1503.01070, 2015. URL http://arxiv.org/abs/1503.01070.

[149] Hugo Touvron, Thibaut Lavril, and et al. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

[150] Hugo Touvron, Louis Martin, and et al. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[151] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014. URL http://arxiv.org/abs/1412.0767.

[152] Yunbin Tu, Xishan Zhang, Bingtao Liu, and Chenggang Yan. Video description with spatial-temporal attention. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, pages 1014–1022, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4906-2. doi: 10.1145/3123266.3123354. URL http://doi.acm.org/10.1145/3123266.3123354.

[153] Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan, October–November 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-8643. URL https://aclanthology.org/W19-8643.

[154] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

[155] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. *CoRR*, abs/1411.5726, 2014. URL http://arxiv.org/abs/1411.5726.

[156] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J. Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *CoRR*, abs/1412.4729, 2014. URL http://arxiv.org/abs/1412.4729.

[157] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence - video to text. *CoRR*, abs/1505.00487, 2015. URL http://arxiv.org/abs/1505.00487.

[158] Subhashini Venugopalan, Lisa Anne Hendricks, Raymond J. Mooney, and Kate Saenko. Improving lstm-based video description with linguistic knowledge mined from text. *CoRR*, abs/1604.01729, 2016. URL http://arxiv.org/abs/1604.01729.

[159] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. volume 1, pages I–511, 02 2001. ISBN 0-7695-1272-0. doi: 10.1109/CVPR.2001.990517.

[160] Jiawei Wang, Liping Yuan, and Yuchen Zhang. Tarsier: Recipes for training and evaluating large video description models, 2024. URL https://arxiv.org/abs/2407.00634.

[161] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *CoRR*, abs/1705.02953, 2017. URL http://arxiv.org/abs/1705.02953.

[162] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. *CoRR*, abs/1711.11135, 2017. URL http://arxiv.org/abs/1711.11135.

[163] Xin Wang, Wenhu Chen, Yuan-Fang Wang, and William Yang Wang. No metrics are perfect: Adversarial reward learning for visual storytelling. *CoRR*, abs/1804.09160, 2018. URL http://arxiv.org/abs/1804.09160.

[164] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Chenting Wang, Guo Chen, Baoqi Pei, Ziang Yan, Rongkun Zheng, Jilan Xu, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2: Scaling foundation models for multimodal video understanding, 2024. URL https://arxiv.org/abs/2403.15377.

[165] Schmitt MR Wen FK. Wang LW, Miller MJ. Assessing readability formula differences with written health information materials: application, results, and recommendations. *Res Social Adm Pharm*, 36: 1179–1187, 2013. doi: 10.1016/j.sapharm.2012.05.009.

[166] Paul J Werbos et al. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[167] Sam Wiseman and Alexander M. Rush. Sequence-to-sequence learning as beam-search optimization. *CoRR*, abs/1606.02960, 2016. URL http://arxiv.org/abs/1606.02960.

[168] BigScience Workshop and et al. Bloom: A 176b-parameter open-access multilingual language model, 2023. URL https://arxiv.org/abs/2211.05100.

[169] Qi Wu, Chunhua Shen, Anton van den Hengel, Lingqiao Liu, and Anthony R. Dick. Image captioning with an intermediate attributes layer. *CoRR*, abs/1506.01144, 2015. URL http://arxiv.org/abs/1506.01144.

[170] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016. URL http://arxiv.org/abs/1611.05431.

[171] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[172] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015. URL http://arxiv.org/abs/1502.03044.

[173] Yang Yang, Jie Zhou, Jiangbo Ai, Yi Bin, Alan Hanjalic, Heng Tao Shen, and Yanli Ji. Video captioning by adversarial lstm. *IEEE Transactions on Image Processing*, 27:5600–5611, 2018.

[174] Yuchen Yang and Yingxuan Duan. Towards holistic language-video representation: the language model-enhanced msr-video to text dataset, 2024. URL https://arxiv.org/abs/2406.13809.

[175] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4507–4515, Washington, DC, USA, 2015. IEEE Computer Society. ISBN 978-1-4673-8391-2. doi: 10.1109/ICCV.2015.512. URL http://dx.doi.org/10.1109/ICCV.2015.512.

[176] Andreas Veit Xun Huang Yin Cui, Guandao Yang and Serge Belongie. Learning to evaluate image captioning. In *CVPR*, 2018.

[177] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. *CoRR*, abs/1510.07712, 2015. URL `http://arxiv.org/abs/1510.07712`.

[178] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. *CoRR*, abs/1609.05473, 2016. URL `http://arxiv.org/abs/1609.05473`.

[179] Youngjae Yu, Jongwook Choi, Yeonhwa Kim, Kyung Yoo, Sang-Hun Lee, and Gunhee Kim. Supervising neural attention models for video captioning by human gaze data. *CoRR*, abs/1707.06029, 2017. URL `http://arxiv.org/abs/1707.06029`.

[180] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016. URL `http://arxiv.org/abs/1605.07146`.

[181] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019. URL `http://arxiv.org/abs/1904.09675`.

[182] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL `https://arxiv.org/abs/2306.05685`.

[183] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. *CoRR*, abs/1804.00819, 2018. URL `http://arxiv.org/abs/1804.00819`.

[184] Pengyuan Zhou, Lin Wang, Zhi Liu, Yanbin Hao, Pan Hui, Sasu Tarkoma, and Jussi Kangasharju. A survey on generative ai and llm for video generation, understanding, and streaming, 2024. URL `https://arxiv.org/abs/2404.16038`.

[185] Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. Streaming dense video captioning, 2024. URL `https://arxiv.org/abs/2404.01297`.

[186] Fangyi Zhu, Jenq-Neng Hwang, Zhanyu Ma, Guang Chen, and Jun Guo. Understanding objects in video: Object-oriented video captioning via structured trajectory and adversarial learning. *IEEE Access*, 8: 169146–169159, 2020. doi: 10.1109/ACCESS.2020.3021857.

[187] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[188] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. ECO: efficient convolutional network for online video understanding. In *ECCV*, 2018.