Learning to evaluate video captioning systems through human assessment

Luis Lebron Casas, MSc

Supervised by Dr. Yvette Graham (School of Computer Science and Statistics, Trinity College Dublin), Dr. Kevin McGuinness and Prof. Noel E. O'Connor

A Dissertation submitted in fulfilment of the requirements for the award of Doctor of Philosophy (Ph.D.)

School of Electronic Engineering

Dublin City University

June 2024

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of PhD is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: Luis Lebron Casas ID No.: 19211684 Date: 14/11/2023

Acknowledgements

This research was supported by the Irish Research Council Enterprise Partnership Scheme together with United Technologies Research Center Ireland and the Insight SFI Research Centre for Data Analytics. This has been a great opportunity to proceed in my career and do a PhD in a stimulant environment.

On a more personal note, I would like to express my gratitude to all my supervisors Dr.Yvette Graham, Dr. Kevin McGuiness, Prof. Noel O'Connor and Dr. Konstantinos Kouramas for their guidance and support. We had to improvise solutions in a period when the world was in chaos. Thanks to their patience and their counselling both in the work and in a more personal aspects this journey has finished on a positive note. Thanks to all of you, I have learned plenty of things from you; not only about technical matters but also, and I think more important, how to present and defend my work to the general public and experts.

I also want to thank the people in Insight SFI Research Centre and ADAPT as I got to know plenty of them and discussed ideas and goals with them. Especially mention my lab mates: Diego, Eric, Enric, Paul, Sen, Jaime and all the others; who were always around to discuss things or go for lunch. Thanks also to Prof. Shin'ichi Satoh and all the people in Shin'ichi Satoh's Lab and NII. It was a wonderful opportunity to spend a few months in your lab and learn from experts working on various video domains.

Finally, my thanks to my parents who have supported me in all these adventures. And my friends who were always there when I needed to disconnect for a moment from research.

Contents

1 Introduction **20**

1.1 Motivation 20

1.2 Hypothesis and Research Questions 24

1.2.1 Hypothesis 24

1.2.2 Research Questions 24

1.3 Structure of the Thesis 26

2 Literature Review **28**

2.1 Introduction 28

2.2 Video Captioning Datasets 28

2.3 Computer Vision, Natural Language Processing and Deep Learning . 33

2.3.1 The Perceptron 33

2.3.2 Convolutional Neural Network (CNN) 34

2.3.3 Recurrent Neural Network (RNN) 35

2.3.4 Long Short Term Memory (LSTM) 36

2.3.5 Transformers 37

2.3.6 Vision Transformers 38

2.3.7 Training the Model's Parameters 38

2.4 Approaches to Video Captioning 40

2.5 Video Captioning Metrics 42

2.5.1 Hand Crafted Metrics 42

2.5.2 Learned Metrics 46

2.6 Datasets for Evaluation 47
2.7 Summary 50

3 A Study of Video Captioning Models in a Real Scenario: Insight@DCU Participation in TRECVid 2019 52

3.1 Introduction 52
3.2 Video Captioning 54
3.3 Model 54
3.4 Dataset 57
3.5 Experimental Results 57
3.5.1 Experiment Setup 58
3.5.2 Replacing the Base LSTM with a Transformer 58
3.5.3 Data Augmentation with Additional Datasets 59
3.5.4 Competition Submitted Set-up 60
3.5.5 Robustness Study: Shuffle Frames 60
3.5.6 Using Additional Datasets 60
3.5.7 Metric Study 61
3.6 Summary 62
3.6.1 Better Evaluation as the First Step to Improve Video Captioning Models 63

4 BERTHA: Video Captioning Evaluation via Transfer-Learned Human Assessment 65

4.1 Introduction 65
4.2 Related Work 68
4.2.1 Video Captioning Metrics 68
4.2.2 Human Judgement 69
4.2.3 Language Models and Transfer Learning 72
4.3 Evaluation of Video Captioning via Transfer Learning 72
4.4 Experimental Results 75

4.4.1 Dataset . . . 75
4.4.2 Human Assessment . . . 76
4.4.3 Evaluation . . . 77
4.4.4 Raw Scores vs Standardized Scores . . . 79
4.4.5 System-level Evaluation . . . 80
4.4.6 Caption-level Evaluation . . . 83
4.4.7 Qualitative Analysis . . . 89
4.4.8 BERTHA as a loss function . . . 90
4.5 Conclusions . . . 92

5 Improving the model using Contrastive Learning 94
5.1 Introduction . . . 94
5.2 Related work . . . 95
5.3 Methodology . . . 98
5.4 Results . . . 100
5.5 Conclusions . . . 103

6 Evaluation of Video Captions Automatically Generated Using Vision and Language Models 106
6.1 Introduction . . . 106
6.2 Related work . . . 108
6.3 Proposed Method . . . 109
6.3.1 Vision and Language Models . . . 109
6.3.2 Evaluation Methodology with a Single Image . . . 110
6.3.3 Evaluation Methodology for a Video . . . 111
6.3.4 Training Models . . . 112
6.3.5 Filtering Poor Human Judgements . . . 113
6.4 Experimental Results . . . 113
6.4.1 Image Captioning Evaluation . . . 113
6.4.2 Video Captioning Evaluation . . . 115

6.4.3 Training Results . . . 116
6.4.4 Filtering Human Annotations . . . 117
6.5 Conclusion . . . 119

7 Learning to Evaluate Video Captioning from Edits and Questions 121
7.1 Introduction . . . 121
7.2 Related Work . . . 122
7.2.1 Human Judgments Datasets . . . 122
7.2.2 Quality Evaluation and Criteria . . . 123
7.2.3 Post-editing evaluation . . . 124
7.3 Proposed Method . . . 125
7.3.1 POES: a New Metric by Post-editing of a Caption . . . 126
7.3.2 Collection Process . . . 129
7.3.3 Validation . . . 132
7.4 Results . . . 135
7.4.1 Overall Score Distribution for the Five Evaluation Criteria . . . 137
7.4.2 Post-editing Captions to Measure Caption Quality . . . 139
7.4.3 Classical Metrics . . . 141
7.4.4 Choosing the Best Combination for POES . . . 142
7.4.5 Automatic Sentences Correction . . . 143
7.5 Conclusion . . . 144

8 Conclusions and Further Work 146
8.1 Thesis Overview . . . 146
8.2 Hypothesis and Research Questions Revisited . . . 148
8.3 Research Contributions . . . 150
8.4 Proposal for Future Work . . . 151
8.5 Closing Remarks . . . 152

A Datasets with Human Judgments for Video Captioning 154
A.1 TRECVid Video To Text Original Evaluation . . . 154

A.1.1 Machine Translation Data 155
A.2 2015 MSCOCO Image Caption Challenge 156
A.3 Composite 156
A.4 Flickr8k-Expert 157
A.5 New pipeline TRECVid Video To Text Evaluation 157

List of Figures

1.1 Example of a video in the TRECVid dataset and associated captions. [Awad et al., 2019] 21

2.1 Examples of videos and corresponding reference captions for some of the most popular datasets. These few examples illustrate the differences inherent in the structure of each dataset. 30

2.2 Example of the basic layers of a Convolutinal Neural Network (CNN). The input layer, consists of the original image. This is followed by a stack of convolutional layers and pooling layers. It usually ends with a fully-connected layer which produces the final output. 34

2.3 Example of a recurrent neural network (RNN). x denotes the input sequence represented by its instances t. h represent the states of the RNN which connects to different outputs: a single value c, another sequences y or a shifted sequence z. 36

2.4 General scheme of TIGEr – image taken from [Jiang et al., 2019]. The first part describes how the model computes the grounding vector between the input sentences and the image. The second part is a visual representation of how to compute the similarity between the rankings of the regions of the image and the similarity between the two vectors. 46

3.1 Example of a video from ActivityNet dataset [Caba Heilbron et al., 2015] with some examples of shots represents by a single image with a highlighted system generated sentence describing the scene. 52

3.2 Schematic representation of the model. The video is divided into segments; these segments are converted into feature vectors using the C3D model. The final vectors are the input to the encoder-decoder architecture which is in charge of predicting the final caption. 55

4.1 An automatic metric (CIDEr in this case) fails to evaluate the caption as a human would. Note that the human assessment scores are both low as the range is between 0 and 100. 66

4.2 Evaluation with the proposed BERTHA video captioning evaluation method. 73

4.3 Score distribution of Video Captioning Caption-level Metric scores with human assessment for both datasets 78

4.4 William significance test of Video Captioning System-level Metric scores with human assessment for TRECVid 2016 to 2020 participating systems. The p-value in a cell means the significance of the row vs the column in the absolute Pearson correlation. 82

4.5 William significance test of Video Captioning Caption-level Metric scores with human assessment for TRECVid 2016 to 2020 participating systems. The p-value in a cell means the significance of the row vs the column in the absolute Pearson correlation. 85

4.6 Word cloud for TRECVid 2020 MA for the references level 91

5.1 Representation of the evaluation of a system-generated caption against a human reference. The same model, BERTHA, is used to predict both embed vectors, x and x'. The distances, e.g. cosine distance between the two vectors, determine how similar are the two captions. A closer caption means that is more similar than a further one. . . . 98

5.2 Example of loss in the training cycle when only using the reference captions vs using data augmentation in the system and human captions. 102

5.3 Example of loss and prediction in the training cycle when using METEOR as a guide for Contrastive Learning. 104

6.1 Example of a video caption removed by the filtering. The middle frame with the caption and its scores are on the left. There are three examples of similar video-caption pairs on the right that have different scores. The words we consider incorrect are represented in red and in green are the ones that the captions get correct, but the outlier does not. 118

7.1 Density plot of TRECVid 2019 human evaluation ratings without quality control sentence 128

7.2 Example of the user interfaces used for collecting the annotations. . 131

7.3 Example of the graphic interfaces used for validating the annotations. This is the initial part where we can observe the different alerts and initially view some initial data like the time it took the annotator to complete the task, their comment and some small statistics of their answer. 133

7.4 Example of the graphic interfaces used for validating the annotations. This is the view in more detail here we can see the captions that the user has corrected, highlighted in green, and the scores they gave to the original sentence. 134

7.5 Difference in the most used words in the system generated caption compared to the post edited captions. 136

7.6 Density plot of TRECVid 2020 human evaluation ratings with and without quality control sentence; squares highlight the main divergences. 138

7.7 Histogram of similarity between sentences that have been modified. Zero means completely different, and one that is the same sentence. . 143

List of Tables

2.1 Characteristics of the most used datasets for video captioning. 32

3.1 Base model and transformer model trained without the TGIF dataset. 58

3.2 Base model (IVTT-BLSTM), image caption model(Image captioning) and base model with image caption text as input (IVTT-BLSTM-Text). 59

3.3 Final results of the model submitted to TRECVid 2019. 60

3.4 Results on the TRECVid 2018 dataset when shuffling the input frames. 61

3.5 Results on the TRECVid 2018 dataset adding different datasets. . . . 61

3.6 Correlation between metrics on the TRECVid 2019 [Awad et al., 2019]. 61

4.1 Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment for TRECVid SA 2019 participating systems; for both the raw score obtain from the annotators and the z score which is the standardize version of the raw one. 80

4.2 Pearson Correlation of Video Captioning System-level Metric scores with human assessment SA for TRECVid 2016 to 2020 participating systems; metric is trained on captions from all other (non-test) years 81

4.3 Pearson Correlation of Video Captioning System-level Metric scores with human assessment MA for TRECVid 2016 to 2020 participating systems; metric is trained on captions from all other (non-test) years 81

4.4 Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment SA for TRECVid 2016 to 2020 participating systems; metric is trained on captions from all other (non-test) years 84

4.5 Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment MA for TRECVid 2016 to 2020 participating systems; metric is trained on captions from all other (non-test) years 84

4.6 Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment MA for TRECVid 2020 participating systems train using a single HUMAN references 87

4.7 Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment MA for TRECVid 2016 to 2020 participating systems permuting the sentences evaluated 87

4.8 Pearson Correlation of Video Captioning and Machine Translation (MT) Caption-level Metric scores with human assessment for TRECVid SA 2016 to 2019 participating systems; VTT (TRECVid Video To Text) is only using the original data; MT is using only data from WMT [Ma et al., 2019]; VTT+MT:VTT Test set is train in all the data from both sets but test only in data coming from VTT; VTT+MT:MT Test set same train as the previous set up but test in only in MT data; VTT+MT:Mix Test set same training as the previous sets and test in a mixture of both data. 88

4.9 Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment for TRECVid SA 2018 participating systems; showing four configurations; No human at all, human captions as a system in both train and test, only in test and finally only in train. . 89

4.10 Comparison of training [Zheng et al., 2020a] using CIDEr vs BERTHA. The stats use to compare are: initial loss, final loss and final results in training. 92

5.1 TRECVid baseline and data augmentation. The metric is Pearson Correlation with human annotations at the caption level. CL refers to basic Contrastive Learning, aug is data augmentation, and ref-aug is referencing only data augmentation 102

5.2 MSCOCO baseline and data augmentation. Metric is Pearson Correlation with human annotations at the system level. The definitions of Mx can be found in Section 5.2. 103

6.1 Flickr8k and Composite Kendall's correlation with human assessment. TRECVid-VTT 2019 and 2020 Pearson's correlation with human assessment. Only using the middle frame to simulate an image captioning dataset. *results from [Hessel et al., 2021] for Flickr-8k and Composite and from [Lebron et al., 2022] for TRECVid-VTT 2019 and 2020. 114

6.2 TRECVid-VTT Kendall's τ correlation with human assessment using frame data. Images is with a single image; similarity is based on computing an average between the frames 115

6.3 TRECVid training using TRECVid-VTT 2016 to 2019 as train/validation and test in 2020. 116

6.4 TRECVid filtering captions based on linearity in the scores and on linearity in the ranks. Results use Kendall's correlation with human assessment and the percentage increase with respect to the baseline. Also shown are the number of image-caption pairs removed and the percentage that this represents of the total number of samples. . . . 119

7.1 Kendall's tau correlation between each individual question between each other in TRECVid 2020. Post editing score (POES) is only applied in the captions that have edits while the others are done in the full dataset. 135

7.2 TRECVid-VTT 2019 Kendall's correlation with human assessment. The human judgements are computed as the mean of all the annotators. 135
7.3 TRECVid-VTT 2020 Kendall's correlation with human assessment. The human judgments are computed as the mean of all the annotators. 137
7.4 TRECVid-VTT 2019 and 2020 Kendall's correlation of POES vs all the other metrics. 140
7.5 Kendall's tau correlation at system level for TRECVid 2019. 10 real systems were increased to 100 "pseudo-systems". 141
7.6 Kendall's tau correlation at the system level for TRECVid 2020. 6 real systems were increased to 100 pseudo-systems. 141
7.7 Kendall's tau correlation of the post editing scores in TRECVid 2020. It contains the editing based on character and word differences with respect to both the system-generated caption (original) and the edited caption. 142
7.8 Study of postediting words replacing them by synonyms and antonyms in the system-generated captions of TRECVid-VTT 2020. The first column refers to the original metrics; the second column shows results using only the CLIP score as a reference to select the sentence; the last column uses the combination of the normalised metrics as a reference. 143

A.1 Stadistics of the human judgments datasets used in this thesis. 154
A.2 Summary of the subset of the TRECVid-VTT dataset use in various experiments across the thesis. 155

Acronyms and Abbreviations

NN ..... Neural Network

CNN ..... Convolutional Neural Network

DNN ..... Deep Neural Network

NLP ..... Natural Language Processing

NLG ..... Natural Language Generation

CNN ..... Convolutional Neural Network

LSTM ..... Long Short-Term Memory

MLP ..... Multi-Layer Perceptron

MSE ..... Mean Squared Error

PCA ..... Principal Component Analysis

BERT ..... Bidirectional Encoder Representation from Transformers

BILSTM ..... Bidirectional Long Short-Term Memory

CLIP ..... Contrastive Language-Image Pre-Training

Publications

Papers published:

- Lebron, Luis; McGuinness, Kevin, and Noel E. O'Connor. "INSIGHT@ DCU TRECVid 2019: Video to Text." TRECVid. 2019.

- Lebron, Luis; Y Graham, Yvette; O'Connor. Noel E and McGuinness, Kevin. "BERTHA: Video Captioning Evaluation Via Transfer-Learned Human Assessment." LREC 2022: 1566-1575

- Luis Lebron, Yvette Graham, Noel E. O'Connor, Kevin McGuinness. "Evaluation of Automatically Generated Video Captions Using Vision and Language Models." ICIP 2022: 2416-2420

Paper to be published:

- Lebron, Luis; Graham, Yvette; O'Connor. Noel E and McGuinness, Kevin. "Learning to Evaluate Video Captioning from Edits and Questions."

Learning to evaluate video captioning systems through human assessment

Luis Lebron Casas

Abstract

Multimodal content analysis has attracted the attention of numerous researchers in the computer vision community. One of the most representative tasks of this sub-field is video captioning. There are numerous difficulties involved in creating these descriptions, ranging from effectively conveying the essence of the scene to generating text that is grammatically accurate and flows smoothly.

In this thesis, we examine an inherent issue with these techniques. The question being asked is *"How can we effectively evaluate video captioning?"*. Our research has identified several shortcomings in the metrics, such as skewness in the length of the sentences or the practice of only measuring textual similarity on a small set of human reference captions. The widely used TRECVid video-to-text task is used as the basis for our investigations. Shortcomings are identified by comparing the correlation of various metrics against the direct assessment. To improve the quality of the evaluation, we propose to fine-tune a large language model to maximise the correlation. Our results show that this metric uses other qualities to evaluate the system output and obtains good performance. This idea is then extended using contrastive learning to learn an embedding space where a more human-like similarity can be used in the evaluation. We show that vision and language models can be used to measure the similarity between visual and textual features.

We also identified discrepancies in how human judgment scores were distributed. Some experiments on reducing bias by collecting multiple scores per caption or filtering outliers were carried out. These finally motivated us to create a new dataset for video captioning evaluation. This new dataset divides the qualitative scores into five sub-aspects and the captions were also post-edited. The final dataset provides the basis for a more comprehensive and robust reference upon which to compare metrics for video captioning evaluation.

Chapter 1

Introduction

1.1 Motivation

Video captioning is the task of describing a video through textual information. It has multiple applications, from developing training video guides to summarising video data using text. Video captioning is an interesting research topic as it combines two primary domains of machine learning: computer vision and natural language processing. A similar task is image captioning, where the descriptions need only explain a single still image. As might be expected, image captioning is a relatively more straightforward task than video captioning. Video needs to consider additional challenges. The first is that video is a data source that consists of multiple still images put together, so the first challenge comes from understanding the relationships that link the frames. Commonly a video will be divided into groups of frames called shots or clips. This normally represents a sub-scene happening between two transitions or can be defined as independent from the other group of frames; for instance, consider a football match – a shot might focus only on the moment from a player kicking the ball to scoring a goal. The relationship between frames then can be expressed through the motion in the scene, identifying the actions happening in the shot, the causality in the image, or the occlusions. Compared to image captioning, the amount of information to be understood is larger, which adds complexity to the methods needed to solve this problem. Whilst audio can also be included as an

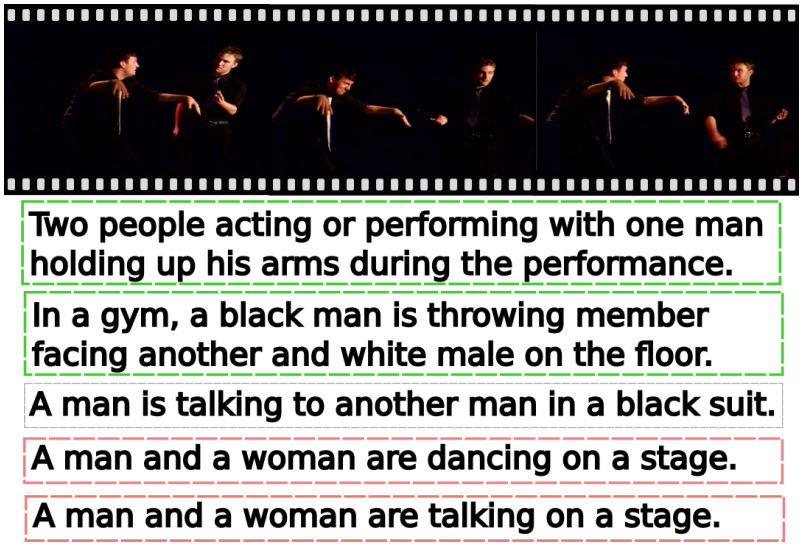


Figure 1.1: Example of a video in the TRECVid dataset and associated captions. [Awad et al., 2019]

input for captioning, it is not commonly used in the literature and is outside the scope of the work reported here. This input adds a high level of complexity to the overall pipeline, however, the results don't show a great improvement in comparison. Moreover, not all the videos in the public dataset contain an audio track[Iashin and Rahtu, 2020b].

Generating free text captions from videos has been made possible largely thanks to deep learning models. Usually, the models are based on the encoder-decoder architecture [Li et al.,]. In the literature, the most common combination is the use of Convolutional Neural Networks (CNN) to extract the visual and temporal features and Long Short Term Memory (LSTM) to generate the final output. Nowadays, newer approaches make use of Transformers and Vision and Language Models [Moctezuma et al., 2022]. More detailed descriptions of both these techniques is presented in the next Chapter 2.

In terms of performance, based on the metrics used in the largest and most challenging datasets, state-of-the-art approaches in the literature still need to achieve significantly higher accuracy. The captions obtained from a typical automated captioning system still differ substantially in many cases from those a human would produce. Although human captions have some major differences and are a biased

benchmark; in this field, they are still considered the golden standard of best captions for a given video. Figure 1.1 shows an example of such inaccurate captions. In this figure, some example system-generated and human-generated captions are displayed. In this example, we can observe many points that make the captions difficult to evaluate. For example, the two first captions are correct, and the last two are incorrect. However, depending on the evaluation criteria used as it is fairly general, the one in the grey box could be either accurate or inaccurate. This criteria is difficult to define in a general way. Another problem is when humans add information to the caption extracted from deductions or external knowledge. In this example, the second caption talks about a gym, which is not obvious from the original video. Another typical example is when a celebrity's name is mentioned when describing a movie clip. The final and main difficulty is to express this qualitative analysis numerically to "explain" the errors to a model and thus have a quantitative basis to judge the caption.

Clearly, the set of textual descriptions that can be accepted as good are naturally very diverse. However, a minimum level of overlap is expected across captions. The level of overlap, between humans and the generated captions, and the diverse nature of this overlap makes it challenging to automatically evaluate captioning models.

Many factors can influence the performance of captioning models, but one of the most relevant is the metric used to study a model's performance. A good metric helps to understand the model and can also help to improve it. As mentioned before, the captions can be quite diverse, so metrics need to capture this diversity. In video captioning, however, the disadvantage is that generally speaking, a single metric typically reflects only one aspect of the model performance appropriately. Hence, multiple metrics need to be used at the same time. Ideally, a good metric should measure numerous aspects of a sentence; fluency, accuracy in the content, and, if available, similarity to the human references. We should be able to differentiate a good metric from a bad one; however, this is challenging. There are also difficulties in representing the semantic relationship of words as well as describing the interactions

of the objects in a natural way. The most challenging aspect is ensuring that the metric adequately reflects human variance; two individuals typically do not produce the same caption for the same video. As such, the best way to evaluate a system-generated caption is to compare it against a reference, as the human will consider all the factors described previously. A good metric is thus usually defined as one that correlates well with these human judgments of caption quality. This establishes two separate sets of human-in-loop instances. First, we have the people in charge of providing the reference sentences that can be used to compare to the system-generated phrases or to train the model. The second instance is the other set of annotators with the task of evaluating the system outputs.

As mentioned before, in the literature the most common way to evaluate a captioning system is to compare the generated sentences against those provided by a human. The usual procedure is to report a set of metrics as each one has its own shortcomings and strengths. These metrics are BLEU [Papineni et al., 2002], ROUGE [Lin, 2004], METEOR [Banerjee and Lavie, 2005], and CIDEr [Vedantam et al., 2015]. The main issue with these metrics is that they are all based solely on textual information provided by a limited set of annotators and do not take into account the visual or audio data present in the original video. They are defined and explained in the next Chapter with a focus on their shortcoming and the new alternatives presented in the literature.

This research focuses on improving video captioning models using better metrics. The research initially studies a specific and popular challenge in video captioning – the TRECVid video-to-text task [Awad et al., 2019]. A model was submitted to the 2019 edition. A study of its weaknesses and how to improve it found that, generally, the metrics used correlate poorly with human evaluations at the caption level. Using different metrics could produce drastic changes in the results independently of the architecture of the captioning system. Based on this finding, research subsequently focused on creating a good metric. This thesis presents the various methodologies explored to improve the correlation with human annotators and to understand how

humans evaluated those correlations.

1.2 Hypothesis and Research Questions

This research generally focuses on improving a video captioning model without focusing on specific methodologies or architectures. The aim is to provide a method that can help all models. From comprehensive research, both from the literature and our own experiments, the metrics currently being used to evaluate performance were identified as the key consideration in this regard.

1.2.1 Hypothesis

The hypothesis underpinning the research reported in this thesis is:

> Video captioning models require enhanced metrics that better correlate with human evaluations of the same captions. As human references are used as the baseline comparison, increasing this correlation can improve both the quantitative results and the sentences produced by the models. It is hypothesised that a learned metric can include the main factors considered in a human evaluation without the need for these to be individually specified. Designing a metric in this way could enable researchers to evaluate video captioning systems independently of the architecture used and select the best instances.

1.2.2 Research Questions

This hypothesis can be split into the following research questions (RQ), which guide the research reported in this document:

- **RQ1** In the most challenging datasets in the literature, none of the models achieve satisfactory results compared to other similar tasks such as image captioning or machine translation. For instance, in a popular dataset for image

captioning like COCO [Lin et al., 2014], one of the top performing systems, CoCa [Yu et al., 2022], achieves a very low score of 33.9 using the METEOR metric. One of the top performing models, VLTinT [Yamazaki et al., 2022], in a challenging dataset for video captioning, ActivityNet [Krishna et al., 2017], only achieved a METEOR score of 17.97. Later in this thesis, the METEOR score will be clearly defined so that the reader can better understand why this is a low score. **This leads to the general question: What are the differences that make video captioning models perform so low compared to other tasks, like image captioning or machine translation[Ma et al., 2019, Vinyals et al., 2016a]?**

- **RQ2** Finding a good metric to evaluate the training and test performance can greatly influence the model output independent of the architecture and the training dataset. This was proven with the runs of [Mantecón et al., 2019], one of the systems participating in TRECVid-VTT [Awad et al., 2020]. In order to evaluate metrics, they are usually compared to human judgments. Some studies show that the performance of the metrics used can be improved, and new metrics can be designed for video captioning [Hessel et al., 2021]. However, designing new metrics is difficult as it is difficult to define what is relevant in the metric. These considerations lead to the following question: **Can a metric learned from human evaluations better correlate with these same human judgments than an engineered metric?**

- **RQ3** Another problem with commonly used metrics is that they only use textual information and discard all the visual data typical of video captioning. There are already proposed metrics that try to join these two domains; however, they are not widely explored [Jiang et al., 2019], typically because of problems related to accessibility, computation cost, etc. This prompts the question: **Can we find a way to combine visual and textual information to evaluate the sentences used for captioning?**

- **RQ4** In other fields like machine translation, there is a comprehensive study trying to capture and separate all the necessary linguistic concepts, e.g. accuracy, grammar, usability, etc., that influence the final evaluation of a translation [Howcroft et al.,]. These have only recently started to be considered for image captioning evaluation, which has produced more questions about what is relevant in a metric. Thus a further research question is: **Can a single metric cover all the relevant aspects of video caption evaluation, removing the requirement to report multiple tests or metrics?**

1.3 Structure of the Thesis

This thesis is structured as follows. In the next chapter, Chapter 2, we describe some key concepts used later in the document. The chapter starts with an exploration of the commonly reported datasets for video captioning. Then we present some general concepts for deep learning and some relevant architectures. After describing these basic models, we describe the different methodologies used in the literature to solve the captioning task. The chapter ends by presenting the two most relevant aspects for the remainder of the research presented: the current family of metrics used for video captioning and the dataset used to evaluate them.

Chapter 3 describes a study to analyse the shortcomings of the video caption model. We presented a general model to the TRECVid video-to-text task [Awad et al., 2019] and analysed the participating systems' results using our model as a baseline. In this way, we identify some general problems and possible improvements. This study concludes that the metrics of the models used for training and model selection greatly influence the final output. The central tenet of this thesis is how to provide better metrics and how to compare them.

In Chapter 4, we report on the development of a better metric using the same TRECVid video-to-text data. In this chapter, we first present a dataset consisting of the output of the participating systems and the resulting score in the direct assessment evaluation. We detect inconsistencies in human judgment and propose

an improved version of the same dataset. With all the data, we explore the finetuning of a large language model to produce a similar evaluation as humans. This model is only text-based, similar to previously used metrics.

Next, Chapter 5 explores the use of contrastive learning to provide a new methodology to produce a learned metric. Although the results don't improve the correlation, they show a promising direction when more data is available. They also prove that a vast number of human judgments are not necessarily needed but that we need enough diversity in the style of evaluations and captions.

The next step in our research is to explore the use of visual information in the evaluation pipeline. In Chapter 6, we demonstrate the application of a vision and language model trained for zero-shot prediction. Specifically, we utilize this model to evaluate video captions. We also present a methodology to filter incorrect human annotations in the judgment dataset.

In the following Chapter 7, we describe how we built a new dataset for video captioning evaluation. The dataset uses the videos and system outputs from TRECVid-VTT 2019 and 2020. Then an additional human evaluation is collected using Amazon Mturk. This dataset divides the evaluation criteria of the direct assessment into five aspects; quality/naturalness, informativeness, clarity, usability, and the presence of the main subject. Additionally, the annotators are asked to edit the caption if they think it is necessary. To use this post-editing score, we present a simple way of converting them to a numerical value, POES. We then compare previously used metrics with the new ones in this new set-up. In our experiments, we archive on-par results compared to the other metrics. This proves that even a simple method to convert the post-edit sentences to scores can obtain good correlations.

The final Chapter 8 gives a summary of the research conducted in this thesis. The results and findings are discussed by relating them to the research questions and hypothesis presented in this Chapter. We conclude with possible future work and some general remarks.

Chapter 2

Literature Review

2.1 Introduction

This chapter presents an overview of the relevant literature for video captioning and metrics used to evaluate captioning systems. This literature review covers the following points: datasets, typical models and architectures used for captioning, general metrics and metrics specifically designed for the task. The objective is to provide the reader with a general understanding and summary of the topics covered in this thesis as well as details of related work. Each chapter thereafter presents a detailed state-of-the-art literature review relevant to the research presented.

2.2 Video Captioning Datasets

One of the most relevant factors that has made it possible to train and use video captioning models is the creation of big datasets. Datasets are needed to train the models and validate them in scenarios similar to what we can expect in a real-world application. There are many datasets, from those targeting particular topics to more generic ones.

Early datasets were focused on a specific task. For example, some of these were focused on cooking like YouCook [Das et al., 2013] or the Max Plank Institute for Informatics (MP-II) Cooking Dataset [Rohrbach et al., 2015]. The videos in these

datasets have a limited range of activities and few changes of scene. For instance, the same background can be repeated across multiple videos. These datasets often consists of long videos divided into medium-length clips, from a few minutes to half-hour segments. The captions are normally short instructions in an imperative form. They are not direct transcriptions as the video also speaks of additional information.

The next set of datasets of note is focused on more generic topics like videos from movies. A dataset like the Montreal Video Annotation Dataset (M-VAD) [Torabi et al., 2015] consists of short clips of a few seconds taken from films. There is no restriction on the activities depicted in the video in this dataset, and the focus is more on generic scenes.

The most extensive set of datasets to be aware of are those that obtain their videos from multiple sources on the internet. Some popular ones are Microsoft Video Description (MSVD) [Chen and Dolan, 2011], and MSR-Video to text (MSR-VTT) [Xu et al., 2016]; both consist of short clips with multiple references. Both contain numerous open-domain videos divided into short clips of around 20 seconds duration. The main difference with other types of datasets is that they have multiple captions annotated in various languages. MSR-VTT is one of the most extensive datasets in terms of the number of videos, with approximately 10k videos. In a similar vein, VATEX [Shi et al., 2022a] consists of 41,269 videos collected from Kinetics-600 and YouTube videos of human activities.

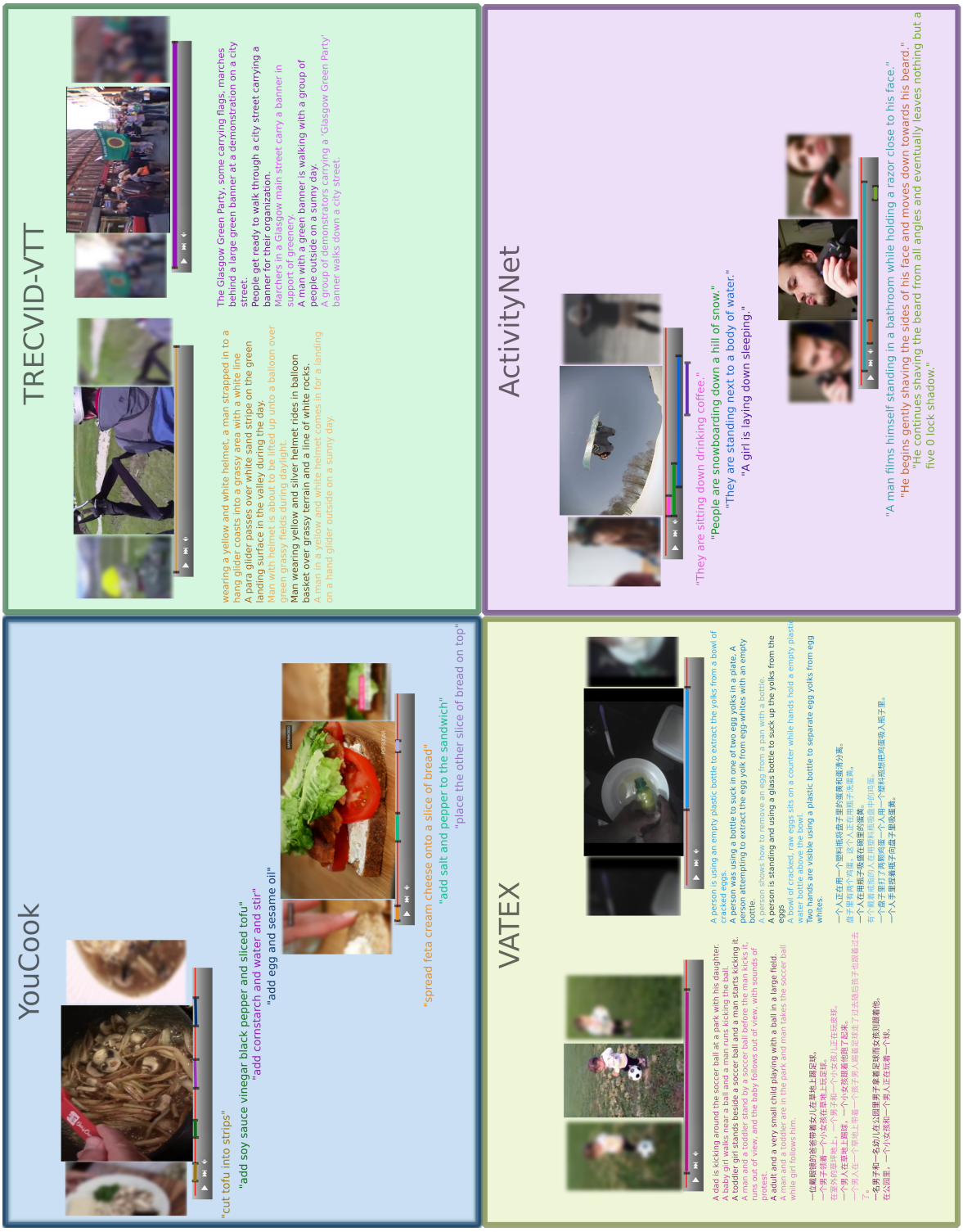


Figure 2.1: Examples of videos and corresponding reference captions for some of the most popular datasets. These few examples illustrate the differences inherent in the structure of each dataset.

The most challenging datasets are focused on dense video captioning. These datasets contain minutes-long videos, and the task is to segment these into clips and produce a caption per clip. ActivityNet [Caba Heilbron et al., 2015] is an example of this type of dataset, containing 100k human descriptions for 20k videos, which cover around 30 seconds of the videos. Unlike other datasets, the annotations in this dataset can overlap, which adds another level of complexity. It is also used for tasks like activity recognition or entity object localisation. ActivityNet has also been used in the video captioning field for multiple sub-datasets and sub-tasks. For example, a subset of the videos was used for object-oriented captions, where the captions are annotated to make explicit references to the objects in the scene [Zhou et al., 2019].

Other datasets come from competitions organised around community benchmarking.The TREC Video Retrieval Evaluation (TRECVid) [Awad et al., 2017] workshops are one such competition. Each year TRECVid provides around 2,000 videos annotated by 2 to 5 human annotators. The organisers later release the videos and annotations, building an extensive dataset over time. The videos are around 10 seconds long. Apart from the standard metrics, TRECVid also uses Direct Assessment [Graham et al., 2018] and text similarity to evaluate the captions generated by participants. Direct Assessment (DA) is the particular methodology to evaluate the captions generated by using a human annotator which is task to give a numerical value of how well the caption describes the video or how similar is the caption to the human written sentences. In the case of TRECVid-VTT this score is between 0 and 100, and the human looks at the system-generated phrase and the original video.

One major problem with all datasets is the availability of the entire dataset. Most videos come from online platforms where the author has little control that the data that will be available for future works or to replicate their own experiments. Due to copyright or the decision of the owner of the videos, some of the videos are taken offline and cannot be recovered. Data protection laws make it even harder to

Dataset	Videos	Captions	Duration	Notes
MSVD	~2000	70028	~10 seconds	Multilingual
MSR-VTT	413000	~8000000	~15 seconds	-
TRECVid-VTT	~2000 each year	~10000 each year	~10 seconds	-
TGIF	102,068	~500000	Up to 6 seconds	-
YouCook	2000	2000	~5.26 minutes	Captions are sequencial
MP-II Cooking	16145	52478	a few minutes	185 videos divided into clips
M-VAD	92	55904	few minutes	Clips from movies
ActivityNet	20000	100000	minutes	Overlapping
VATEX	41300	8260000	~10 seconds	Multilingual

Table 2.1: Characteristics of the most used datasets for video captioning.

keep a local copy of the video and distribute it. Moreover, some of these datasets are not even available for research, or only partial data is provided. These factors can make it challenging to compare a specific approach to others and build on top of them.

Finally, Table 2.1 show some of the most used datasets with some information about their features. It should be noted that there is no common approach to how they were annotated, so the captions in each can be very different in nature. Some of these datasets have a rigorous procedure for annotating the video, e.g. the number of words, what part to focus on, etc.; others are much more flexible.

The differences between the types of captions available are depicted in figure 2.1. From the figure, it can be seen that the YouCook dataset contains videos focused on cooking, so the captions only describe the steps to prepare the dish and are assumed to be sequential. All the other datasets are composed of videos of different topics, and the captions aim to describe the scene. In TRECVid-VTT [Awad et al., 2017], the caption describes the short video, but it contains up to five human-written sentences per video. The VATEX [Shi et al., 2022a] dataset only represents a short video segment and provides multiple captions; additionally, it includes text both in English and Chinese. The ActivityNet dataset is different to the others as it is aimed at dense video captioning; it captions multiple clips that can overlap.

2.3 Computer Vision, Natural Language Processing and Deep Learning

Most modern problems in the field of machine learning encounter barriers when trying to include all the possible factors influencing a decision. For instance, the shape of an object can change depending on the observer's point of view, or the colour can vary depending on the lighting conditions. Classical handcrafted methods need a human to consider all the possible differences in the data. So, for example, a human needs to design a way to represent all the possible camera angles and colours of the object and tailor these features to the problem the method is solving. Deep learning solves this problem by allowing the model to learn its data representation. The model is trained to choose these features from the available data. Video captioning has significantly benefited from the introduction of deep learning. In this subsection, some of the most common architectures are presented to better understand the concepts covered in further chapters.

2.3.1 The Perceptron

The perception [Minsky and Papert, 1969], or neuron [Michie et al., 1995], is the basis of a neural network and is inspired by a biological neuron. The perceptron is a mathematical function that takes a set of inputs and linearly combines them. This process had a tendency to suffer the vanishing gradient or exploding gradient problem. These problems refer to when the gradient of the function is so large or small that no update can find the desired minimum. Similar to the perception [Michie et al., 1995] proposes the neuron where they add the condition that the input can be any numerical value contrary to binary classification and the output is the result of applying a non-linear function to the internal values of the neuron. The linear process is then composed of weights and a bias, and the non-linear function usually is termed the activation function. These new perceptrons or neurons can be grouped into layers and stacked to form a multilayer perceptron (MLP). These more

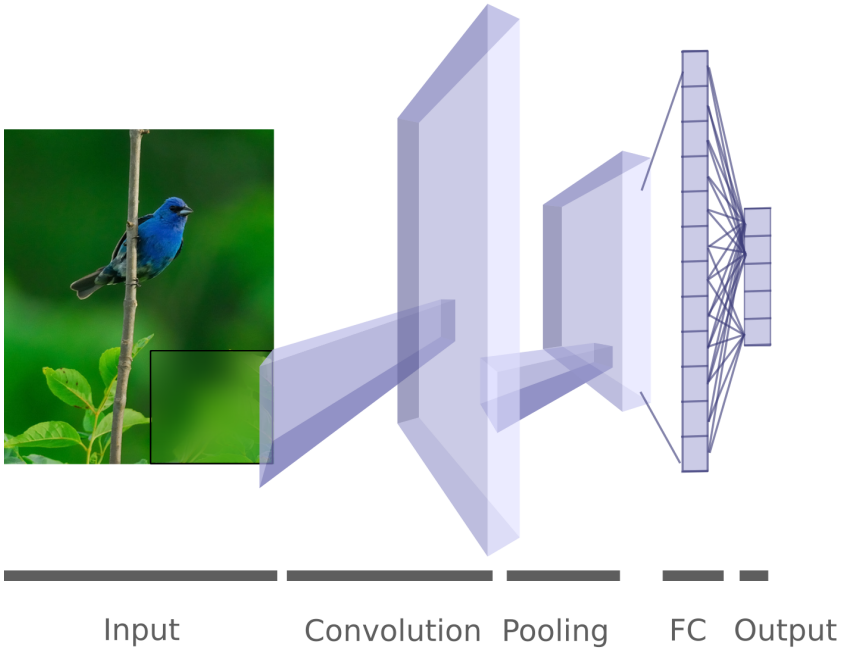


Figure 2.2: Example of the basic layers of a Convolutinal Neural Network (CNN). The input layer, consists of the original image. This is followed by a stack of convolutional layers and pooling layers. It usually ends with a fully-connected layer which produces the final output.

complex models are known as neural networks (NN). The idea behind stacking the perceptrons can be seen in two ways; each layer applies a different mathematical function, so new representations of the data are obtained, or each layer gains more details compared to the previous one, so it can extract more information the deeper it goes. This basic neural network is still in use in many models as the last activation layer or to change the dimension of the feature vector.

2.3.2 Convolutional Neural Network (CNN)

The previous architecture had the problem that it requires many operations and the focus is dispersed across all the activations of the neurons, the knowledge store in the network quickly became redundant. This makes it fairly costly to compute in big inputs like images or long text sequences. Another problem is that it is not invariant to translation or scale. A popular solution in the image and often in the video domain is to use a convolutional neural network (CNN)[LeCun et al., 2015]. A common image input has three dimensions (RGB), and the CNN use stacks of layers

to obtain the desired final result, deeper in the network the number of filters can be different to the initial three. The basic layers are named: convolutional, pooling, and fully connected (FC).

A convolutional layer reduces the number of connections by only showing a small area of the input image or previous layer based on a shared filter in the layer. The idea behind a convolutional layer is to convolve a filter across all the images and a map of the filter's dot product with the images' area. This results in an activation map which represents the filter's response compared to the input image to identify patterns. This layer can be applied to any dimensional input, not only RGB. The size (width and height) of the filters is smaller than the input layer; however, there is no limit to the number of filters used. The depth of the output layer will depend on the number of filters used.

The second layer, the pooling layer, uses a filter to select the maximum number in an area of the input. This reduces the size of the feature map for the layer. This process discards information from the network, so it should be performed in steps instead of reducing the dimension in a single layer.

The FC is usually the last CNN layer and is often an MLP. It is used to compute the final representation of the network, for instance, to reduce it to a single number to solve a regression problem.

CNNs are now the base architecture used for most image processing problems. They have worked well in multiple tasks such as image classification [Deng et al., 2009] or image segmentation [Ronneberger et al., 2015]. They are also the backbone for other architectures to extract an initial feature vector, as they have high representative power. CNNs are proven to capture general and local features without human supervision, and this characteristic is independent of the task.

2.3.3 Recurrent Neural Network (RNN)

A famous architecture in video and text for sequential problems is a recurrent neural network (RNN) [Rumelhart et al., 1986]. A CNN sees the input as a grid, while an

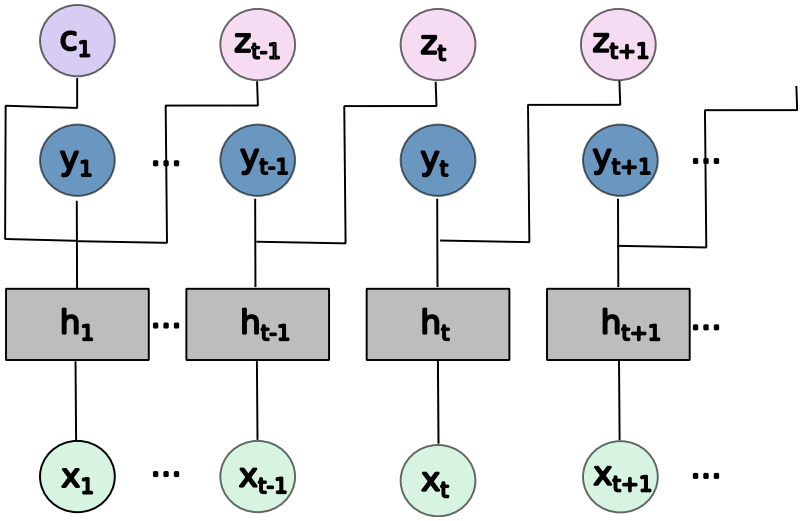


Figure 2.3: Example of a recurrent neural network (RNN). x denotes the input sequence represented by its instances t. h represent the states of the RNN which connects to different outputs: a single value c, another sequences y or a shifted sequence z.

RNN sees it as a sequence. This makes RNN work best in sequence predictions like time series where the CNN are not able to capture the temporal domain easly. RNNs are based on a hidden state that is updated along with the input sequences. These hidden states are shared across the network and will behave differently depending on the change in each update. Its neuron is a function that takes the current state, the previous state and the current input to generate the output. Figure 2.3 demonstrates how this architecture works on sequences. These architectures help to represent a video's temporal information and convey the spatial information of a sentence.

2.3.4 Long Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] networks are a special kind of RNN. This architecture aims to avoid the problem of forgetting earlier data over long periods, which is a common problem of RNNs. The central idea of an LSTM is to use a gate system to regulate the amount of information kept and which one to modify. The first gate, named a forget gate, decides the knowledge to remove from the previous state. Then the input gate chooses the amount of new data to add and send to the next state. The last gate decides the output of the

given cell and what information is fed to the next state. LSTMs have been popular in sequential tasks like video understanding or text summarization.

2.3.5 Transformers

RNN and LSTM, in general, have problems related to processing the time-series sequentially. They have difficulty modelling the dependency in long sequences or the size of the model which is directly linked to the size of the input sequences. Transformers [Vaswani et al., 2017] is a popular architecture used in Natural Language Processing which addresses many of these problems by processing the data in parallel and using self-attention to represent these dependencies. They follow an encoder-decoder structure but don't rely on recurrency like LSTMs or RNNs. The encoder consists of a stack of multi-head self-attention fed to a fully connected feed-forward network. Self-attention in this context refers to the mechanism which allows a single position of the input sequence to interact with the other positions of the same sequences. This stack is applied simultaneously to each input segment, which removes the ordering information. In most cases, this is not desirable, so the transformer includes a mechanism to encode the position. The position encoding is then added to the input embedding. The decoder follows a similar architecture to the encoder. It also consists of N stacked layers. The first layer of the decoder is only applied to the previous part of the sequences, using a mask to hide the other values. The output of this part is then fed to another set of layers with the same structure as the encoder. This sublayer has access to all the sequences. This sublayer's output is fed to a softmax layer to produce the next part of the output sequences. This base architecture is included in multiple state-of-the-art language models—for example, BERT [Devlin et al., 2018], GPT-3 [Brown et al., 2020a] or CLIP [Radford et al., 2021a].

2.3.6 Vision Transformers

A transformer is an architecture designed to be applied over a sequence. However, its excellent performance in various textual tasks has motivated its use in the image domain. The problem is that an image is not a linear sequence; each pixel is related to all the surrounding pixels. To solve this, the authors of [Dosovitskiy et al., 2021] propose to divide the image into patches and feed the patches in sequential order to the Transformer model. One of the most popular models where Vision Transformers are applied is CLIP [Radford et al., 2021a].

2.3.7 Training the Model's Parameters

The described models use two kinds of parameters: the hyperparameters, which are defined beforehand, and the common parameters like weights and other filters, which need to be learned in a training phase. [LeCun et al., 2015] To explain the training phase, we need to consider that the optimisation problem in machine learning is based on getting a specific performance in a measure P. In this scenario, the cost function $J(\Theta)$ doesn't exactly represent the measure, but we expect that by minimising the cost function, we maximise the measure. So the cost function can be written as:

$$J(\Theta) = E_{(x,y)\sim\hat{p}_{\text{data}}} l(f(x;\Theta), y) \tag{2.1}$$

where l is the per-example loss function, $f(x;\Theta)$ is the output of the model, $\hat{p}_{\text{data}}$ is the empirical distribution, and y is the target output. It should be noted that we don't normally have the true distribution, so the model is only trained on an empirical distribution. In deep learning, the objective is normally to minimise the empirical risk. This produces two problems; the first is that this process is prone to overfitting on the training data, and the second is that most of the problem-related metrics are not differentiable. Overfitting refers to when the model only learns the training data and is not capable of generalising. To minimise the loss, which is the objective of the model, we need the derivative of the model's function, however in

most cases that function is not differentiable. As such, typically, these models are optimized using a surrogate loss function with early stopping criteria.

The algorithms used to minimise the loss function are typically based on gradient descent (GD). This is an iterative process where GD explore the surrounding area in the loss function space to find a minimum. Then the internal parameters are updated, and the cycle is repeated until the early stopping criteria is reached.

The formula for GD is normally expressed as:

$$\Theta_{t+1} = \Theta_t - \alpha \nabla_\Theta l(\Theta_t), \tag{2.2}$$

where t is the current iteration in the GD, α is a hyperparameter called the learning rate (in some variations, it changes during the training phase), and $\nabla_\Theta l$ is the derivative of the loss function in the current instance. When the NN has multiple layers, the chain rule is applied to compute the gradient; this process is referred to as backpropagation.

This process does have some problems as it needs to be computed over the full dataset. This means that it needs to process all the samples of the dataset in a single pass in order to compute Θ_{t+1}. This has many limitations in terms of scalability and computational requirements. To reduce this drawback, these methods are trained using mini-batch stochastic gradient descent (SGD), where only a few random samples are used in the update step. The size of this mini-batch then becomes a hyperparameter.

In traditional methods, the function is tailored to be convex; however, in deep learning, the function is non-convex. This generates some challenges that need to be taken into account when selecting the data, the hyper parameters and when designing the model. First, SGD can produce an increase in the cost function as it is an approximation of the true gradient and its convergeence path is noisy. Second, the loss function can have an infinite amount of local minimum of high cost, which can cause problems to SGD. Also it can have plateaus which means that the gradient is near zero producing no progress in the training phase. Finally, RNNs in particular,

have a problem with vanishing or exploding gradients wherein the update phase, the gradient became extremely large or small.

2.4 Approaches to Video Captioning

Video captioning models can be split in two phases: the first one consists of extracting some information from the video, and the second one is generating a textual description using this data. Additionally, in the task of dense video captioning, a pre-task is how to segment the video such that an approach can produce meaningful descriptions[Chen et al.,].

The initial solution proposed for the problem of producing video captions was template based. The idea is to focus only on the video's main points and insert the words in a predesigned template. The initial models only focus on using language to describe a specific action, like using verbs to describe the movement of cars [Koller et al., 1991] or find the subject, verb and object (SVO) of the scene [Kojima et al., 2002]. These models are relatively simple, but this concept of SVO helped guide the development of future video captioning solutions. The first subject that captured the interest of researchers was people; however, with better-handcrafted detectors, researchers could include more objects, expanding the vocabulary [Hanckmann et al., 2012]—for example, using Bayesian Classifier Binarization to find the image descriptors and then using a Hidden Markov Model to obtain the final output [Das et al., 2013]. From there, the natural progression was to tackle open domain models, and thanks to better datasets and better techniques, the number of entities considered could be significantly increased [Krishnamoorthy et al., 2013, Thomason et al., 2014]. As generating new sentences was very challenging at the time, there was also a family of algorithms that tried to pair videos with pre-generated captions [Sun and Nevatia, 2014]. Algorithms like Random Forests or probabilistic models like Conditional Random Fields [Xu et al., 2015] were used to link visual representations with textual representations.

As deep learning models gained traction in computer vision research, they also

started to be featured in video captioning approaches. The starting point for this was to use a CNN as the phase one algorithm. They were used to detect the SVO and then insert the entities in a predefined sentence [Xu et al., 2015]. However, this approach was subsequently overtaken by an encoder-decoder pipeline to extract features and then produce the final sentences [Yao et al., 2015].

Commonly the feature extraction step is done with a 2D or 3D CNN [Guo et al., 2016], followed by an RNN to generate the text [Hori et al., 2017]. One important feature of video is the motion data between the frames. Some research tries to include it via a 3D CNN only [Pan et al., 2016]. However, other works have also included a specific mechanism to reflect the flow, taking ideas from another field, like action recognition or object detection. This includes ideas like Motion History Images [Zhang and Tian, 2016], using Vector of Locally Aggregated Descriptors (VLAD) features [Hara et al., 2017] to describe the spatial-temporal relation between features or using Region Proposals to limit the focus of the network [Agrawal et al., 2020]. More recently, the models have become more complex in that they try to explicitly lead the DNN into a set of entities such as using an object graph to describe the relationships [Zhang et al., 2020] or with attribute detection and prediction [Huang et al., 2020]. By considering the input as a sequence, not only related by the flow information, some papers try to extract a representation using an RNN. These aim to represent both motion and actions happening in the video [Wang et al., 2020, Pendurkar et al., 2020, Yamazaki et al., a]. Some of the most successful models use either an LSTM [Suin and Rajagopalan, 2020] or a Transformer [Xia et al., 2021, Zhu and Yang, 2020, Yamazaki et al., b].

In terms of the second step, sentence generation, the most popular network to use initially was an LSTM [Yao et al., 2015] as they were the best-performing RNN. Thanks to their flexibility, they continue to be used in multiple tasks and require less resources for short phrases and relatively small datasets than models using Transformers. LSTMs can be improved with mechanisms like attention [Perez-Martin et al., 2021, Yamazaki et al., 2022] or syntax-awareness [Zheng et al., 2020a]. More

recent approaches have also included Transformer models in the decoder [Aggarwal et al., 2020, Iashin and Rahtu, 2020a, Awad et al., 2023, Yamazaki et al., b].

2.5 Video Captioning Metrics

Another relevant part of solving the problem is to actually measure how well we have solved or are solving the challenge. In an ideal scenario, the end-user will be in charge of continuously evaluating the performances of the model, however, that's not feasible. Although the various models can have very different inputs, they are all expected to output a textual description of the video or video clips. For this reason, the standard way to compare these solutions is to evaluate the quality of the sentence produced. To do so, most metrics compare the generated caption against a set of human-written captions for the same video or clip. In the literature, the system-generated caption is also referred to as a candidate caption or hypothesis caption. The human-produced sentences are termed reference captions for evaluation purposes.

2.5.1 Hand Crafted Metrics

BLEU [Papineni et al., 2002] is a machine translation metric commonly used in video captioning. It is based on computing the geometric mean of n-grams match counts. The main idea is to count the number of sequences of characters between a hypothesis caption and a reference caption at different levels (n). This can be understood as counting the precision of the candidate sentence. As such, it depends on the number of matching words between the references and the hypothesised caption. The mathematical formulation of BLEU is the following:

$$\text{BLEU}(c_i, S_i) = \exp\left(\max\left\{0, \frac{r}{d-1}\right\}\right) \cdot \exp\sum_n w_n \ln p_n(c_i, S_i), \tag{2.3}$$

$$p_n(c_i, S_i) = \frac{\sum_{z \in G(c_i)} \min_{s_{ij} \in S_j}(M(z, c_i), M(z, s_{ij}))}{\sum_{z \in G(c_i)} G(z, c_i)}, \tag{2.4}$$

where c_i is the candidate sentences, S_i is the reference sentences, d is the length of the candidate corpus, r is the length of the references corpus, M is the number of substrings shared between the vectors of words, G is the set of n-grams, and w is a weighting vector. The main problem of BLEU is that it penalises long candidates. Apart from only measuring the similarity between the sentences, it depends on the references, and it doesn't evaluate synonyms or valid sentence restructuring. As it is a measure of similarity it is also easy to fool – it is only required to put a few correct words in the right places to get a relatively high score.

ROUGE [Lin, 2004] is similar to BLEU as it uses the overlap of n-grams between the reference sentences and the generated ones. The difference is that ROUGE considers the n-gram occurrences in the total sum of the number of reference sentences, while BLEU considers the occurrences in the sum of candidates. It has similar problems as BLEU, but in this case, it favours longer sentences.

These basic metrics lack the capacity to analyse the sentences and are affected by simple transformations like using synonyms. METEOR [Banerjee and Lavie, 2005] is a more complex metric as it can effectively use the information of multiple references. METEOR reaches exact token matches, stemmed tokens, paraphrase matches and semantically similar matches using WordNet synonyms. The metric is identical to BLEU and ROUGE, but it assigns weights to some of the words depending on whether the word is an exact match, a synonym or a stemmer. It is one of the best-performing metrics in cases of a small number of references. Continuing with the same notation and additionally dividing the candidate and references into content words and functional words, the formula for METEOR can be expressed as:

$$P = \frac{\sum_t w_t \cdot (G'_t(c_i^{\text{content}}) + (1-\delta) \cdot G'_t(c_i^{\text{function}}))}{\delta \cdot |c_i^{\text{content}}| + (1-\delta) \cdot |c_i^{\text{function}}|} \tag{2.5}$$

$$R = \frac{\sum_t w_t \cdot (G'_t(S_i^{\text{content}}) + (1-\delta) \cdot G'_t(S_i^{\text{function}}))}{\delta \cdot |S_i^{\text{content}}| + (1-\delta) \cdot |S_i^{\text{function}}|} \tag{2.6}$$

$$F_{mean} = \frac{P \cdot R}{\alpha \cdot P + (1-\alpha) \cdot R} \tag{2.7}$$

$$Pen = \gamma \left(\frac{ch}{k} \right)^{\beta} \tag{2.8}$$

$$\text{METEOR} = (1 - Pen) \cdot F_{\text{mean}} \tag{2.9}$$

where α, β, γ and w are parameters chosen to maximize correlation. Pen is a fragmentation penalty to consider the change in the word order. G' is a matching function taking into account tokens, stemmed tokens and other operations in WordNet. This solves the problems related to the use different ways to reformulate the same sentences. However, it does not account for issues such as variations in reference captions. It relies solely on a fixed set of references and measures similarity only with them.

The previous metrics always assume that there is a high consensus between the annotators and that their caption always provided all the relevant information in the video. CIDEr [Vedantam et al., 2015] is the most used metric in the current literature as it consideres a certain level of disagreement between the refernces sentences. Initially, it was proposed as an image captioning metric but it potential has make it popular in other problems. CIDEr measures the consensus between system-generated captioning and the set of reference sentences. This brings the advantage that it measures: saliency, importance, accuracy, grammar and reference agreement. Saliency refers to which parts of the sentences are relevant, importance measures how much information that word adds, accuracy is how correct is the use of those words, grammar accounts for the linguistic quality of the sentences and reference agreement takes into account if there are mayor differences between annotators. The formula for a single n-gram is :

$$\mathbf{CIDEr}_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{g^n(c_i) \cdot g^n(s_{i,j})}{\|g^n(c_i)\| \|g^n(s_{i,j})\|}, \tag{2.10}$$

where g^n is the TF-IDF vector. TF-IDF is the term frequency-inverse document frequency, which evaluates the importance of a word in a sentence within a corpus or collection of sentences. CIDEr addresses the possibility that annotators don't include the same information and that a caption could have different levels of detail.

The final metric consists of the sum of all n-grams.

One interesting aspect that also needs to be included in the evaluation of the caption is the scene in itself. The sentences contain semantic propositional content that is interesting to contemplate in order to evaluate the correctness of the sentence. SPICE [Anderson et al., 2016] is a recent metric that uses a scene graph to describe the sentences allowing it to identify similar sentences in a more generic way than the other metrics. Using a scene graph makes it a robust a common approach to game the metrics; for example, the caption's length doesn't influence the evaluation, nor does the use of synonyms. The first main drawback of this metric is that it doesn't evaluate fluency or grammar. Also, this metric is expensive to compute, which is unsuitable for training deep networks even if a loss function is designed around it.

Although the final output needs to be a summary in textual form of this scene, when evaluating a new sentence is interesting to go back and look at the overall scenario and all the details. Like the two previous metrics, TIGEr [Jiang et al., 2019] is also focused on the image description task. However, this metric tries to specifically include the visual information in the evaluation. To do so, it is evaluated at two levels: first, a *text-image grounding* and second, a *grounding vector comparison*. The first stage consists of finding what part of the images the candidate sentences focus on most. This could be understood as trying to find the focus of attention in the image. The final score is the average between the ranking of the regions and the similarity in attention. A visual representation is displayed in Figure 2.4. This metric doesn't generalise well in more complex datasets and it needs to be re-adapted, which it is costly. It also lacks the evaluation of structure, grammar and fluency.

There are two common considerations in the literature when choosing a metric: it can evaluate the textual part of the caption against a set of references or assess if the caption's information is present in the video. We can suppose the last part is the most relevant when analysing a video description; however, as it is difficult to define what data to focus on, the evaluations don't exhibit much improvement

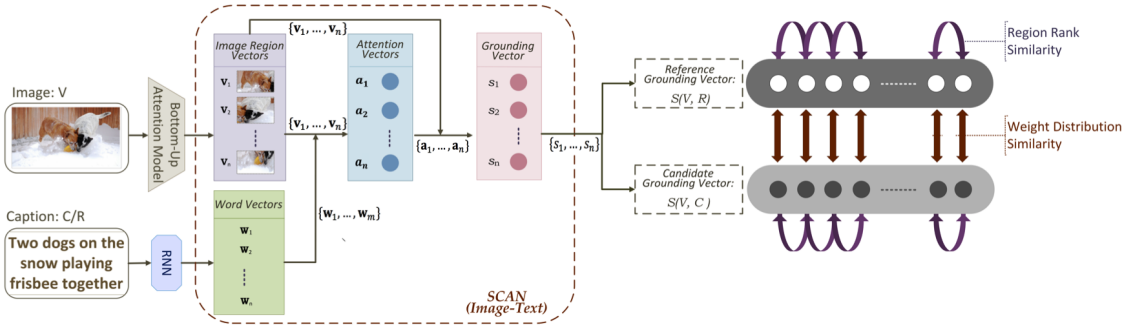


Figure 2.4: General scheme of TIGEr – image taken from [Jiang et al., 2019]. The first part describes how the model computes the grounding vector between the input sentences and the image. The second part is a visual representation of how to compute the similarity between the rankings of the regions of the image and the similarity between the two vectors.

in terms of comparing a particular metric to direct human evaluations or the other metrics [Hessel et al., 2021]. These types of metrics need to understand what a human expects from a caption.

Most of these metrics have a few parameters with default values or that can be tuned to specific datasets which helps to increase the correlation with human judgments [Xu et al., 2019]. Extending this feature, there is a new set of metrics in which the main innovation is that they are trained to perform the evaluation, leaving the decision of what is relevant for the evaluation to the model. These align well with the idea that it is challenging to establish a limited set of criteria for such a task. Some examples of these metrics are presented in the next section. .

2.5.2 Learned Metrics

In domains like machine translation and image captioning, it has become popular to use learned metrics. These are metrics trained to maximize the correlation with human judgements. A task like WMT Metrics Shared task [Mathur et al., 2020] for machine translation and datasets like Microsoft COCO [Chen et al., 2015], or PASCAL [Vedantam et al., 2015] contain the human judgement of the ground truth sentences which can be used to train these metrics. Most of these methods initially used handcrafted features and traditional regression models; however, recently it has become more popular to use language models and other deep-learning methods

to solve this problem.

RUSE is a learned metric that uses three pre-trained sentence embeddings to obtain a final score which ensures a better correlation than the standard metrics for machine translation. BERTScore [Zhang* et al., 2020] is a multi-domain learned metric tested in machine translation and image captioning. It uses a pre-trained language model to compute an embedding and then uses the similarity between the embedding to obtain the final score. In a more simplified setting, some models have aimed only to predict if a caption is generated by humans or by a system. LEIC [Cui et al., 2018] is a trained neural network to classify sentences to determine whether they are human-generated. It obtains state-of-the-art results in the COCO dataset.

These metrics produce better human correlation than classic metrics like SPICE. They are also more robust to adversarial examples and to the pathological transformation [1] of the sentences, which are some of the critical challenges faced by n-gram metrics. More details will be included in later chapters, as the study of these metrics' viability is one of this thesis' main contributions.

2.6 Datasets for Evaluation

To evaluate metrics it is common to use a dataset of the outputs of various systems along with the human judgements of these outputs. There are very few datasets of this kind for video captioning evaluation. As mentioned, most metrics are borrowed from other tasks, so they are usually tested on datasets specific to those tasks.

The most popular task on which the metrics are evaluated is Machine Translation. This is the task where the most extensive study has been performed for Natural Language Generation (NLG). In this field, there are even tasks related to improving the evaluation and obtaining better correlation with human judgments [Mathur et al., 2020]. This kind of initiative has motivated a significant amount of research related to the topic. Looking at the number of articles published only in the International Natural Language Generation Conference (INLG) and the European Work-

[1] Erroneous sentences which are built to get high scores

shop on Natural Language Generation (ENG), from 2000 to 2020, it can be seen that around 200 are related to the human evaluation of systems outputs [Howcroft et al., 2020].

However, these datasets are not suitable as a reference for video captioning evaluation as the input to the systems is different. In translation, the input and output sentences are expected to be similar because both should have the same exact concepts and maybe even retain the same structure. For instance, if concept A is presented before concept B in the original sentence, it is expected to follow the same pattern in the final text. However, this is not the case when the input is images or videos. Concepts can be discarded as needed to be more relevant, the level of detail can also vary, and the structure could be completely different. That is why it is necessary to explore the performance of the metrics in datasets for image captioning. Although fewer in number, the task is more closely related to video captioning.

The most general way to study agreement with human evaluation is to consider this at the system level.[2] If the metric correlates well with the ranking of models, a human usually considers that the metric works well. The most used dataset for this is MSCOCO where the human judgments for the 2015 MSCOCO Image Caption Challenge [Vinyals et al., 2016a] are also provided. The scores provided refer to the following considerations:

M1 Percentage of captions evaluated as better or equal to human caption.

M2 Percentage of captions that pass the Turing Test.

M3 Average correctness of the captions on a scale of 1-5 (incorrect – correct).

M4 Average amount of detail of the captions on a scale of 1-5 (lack of details – very detailed).

M5 Percentage of captions that are similar to human description.

[2]A single aggregated score is used per system, when multiple systems produce the captions. The evaluation then only tries to rank the models not the captions.

However, only 1 and 2 were used to perform the final evaluation[Vinyals et al., 2016a], and the only ones used to test the metrics. The most significant disadvantage of evaluating at this level is the number of systems available and the points used to score performance. In the case of MSCOCO, only six systems participated, so in the end, only 12 data points were used.

A more challenging type of evaluation is at the caption/sentence level. All the systems outputs from all the systems are taken and ranked together[Hessel et al., 2021]. These provide thousands of data points to better identify correlations. One of these datasets used for this is Flickr8k-Expert, which consists of 17000 human judgements on a scale of 1 to 4 points. Flickr8K-CF (from CrowdFlower) is a similar dataset where 48K pairs of images and captions were evaluated with a binary score as the result of the question *"is the caption written by a human?"*, and provided three judgements per pair.

A more extensive dataset is Composite [Hessel et al., 2021], which contains pairs from MSCOCO, Flickr8K and Flickr30K [Vedantam et al., 2015]. The final size is around 4000 pairs with five reference captions. One noteworthy aspect of this dataset is that one of the references is also included in the human judgment process as a system, so it is possible to identify how well it performs compared to human-written captions.

Other different kinds of datasets are Pascal-50S and ABSTRACT-50s [Xu et al., 2019], where the captions are not evaluated to be ranked but distinguished from human-produced sentences. These datasets were proposed together with CIDEr[Vedantam et al., 2015]. The annotation process consists of showing a set of references, and the annotator should pick the closest caption. The sentences are a combination of both sources: human-wroten or machine-generated. And they are paired as human-human, machine-human and machine-machine. They also mix sentences with mistakes.

To the best of our knowledge, there are only two publicly available datasets for video captioning evaluation VATEX-EVAL [Shi et al., 2022a] and TRECVid-

VTT [Lebron et al., 2022]. The first consists of 3,000 videos from VATEX with six captions per video. The captions come from the ground truth, two models and three random captions from other videos in the same dataset. Three English speakers performed the human evaluation and were asked to give a score between 1 to 5 to the set of captions in terms of consistency. The second was developed as part of the work reported in this thesis and is presented in detail in Chapter 4. This dataset comes from TRECVid-VTT[Awad et al., 2019, Awad et al., 2020, Awad et al., 2021]. One of their particularities is that one of the runs of each system is evaluated by direct assessment. As mentioned previously, direct assessment consists of evaluating the system-generated caption by including a human in the loop who gives a numerical score between 0 and 100 on how well the caption describes the video. The process consists of showing the video to a human annotator and only looking at the system-generated caption without references, giving a score between 1 and 100. In our work, reported in [Lebron et al., 2022] we present this dataset and an extension with more annotators per caption to reduce the uncertainty produced by the annotator's bias. This dataset is presented in more detail in Section 4.

We extend the TRECVid dataset by subdividing the general quality score into different questions to reflect other aspects of the caption. The concepts that we evaluate are the following: quality/naturalness, informativeness, clarity, usability and if it clearly displays the main actor/subject of the video. The definition and implication of these terms can be found in Chapter 7. Apart from evaluating the standard way of using scales, in this dataset, we also ask the annotator to edit the caption. This new methodology provides additional information to detect particular errors from a system but it also helps to discover general mistakes done by models. This is explained in detail in Chapter 7.

2.7 Summary

This chapter presents some basic concepts relevant to video captioning that are referred to in later chapters. First, in Section 2.2, a description of the most popular

datasets is presented covering the sources used and whether annotations are available or not. Then in Section, 2.3 covered the basic architecture, methods and training procedures typically used for video captioning, which are mainly based on neural networks. Section 2.4 identifies some of the most relevant works in the field of video captioning. Finally in Section 2.5.2 the main metrics used for evaluating video caption models are described. This is the central topic of this thesis, and more specific details are introduced in the following Chapter which both identifies the main flaws of these methods and proposes an improvement to the evaluation pipeline.

Chapter 3

A Study of Video Captioning Models in a Real Scenario: Insight@DCU Participation in TRECVid 2019

3.1 Introduction

The goal of this chapter is to explore some of the most relevant problems in the video captioning literature. As presented in the previous chapter, there are many styles of datasets for video captioning. However, the architectures used in captioning systems are typically almost identical and yield comparable outcomes. Figure 3.1 shows the quality of the output of a model in one of these datasets where it can be seen that

Figure 3.1: Example of a video from ActivityNet dataset [Caba Heilbron et al., 2015] with some examples of shots represents by a single image with a highlighted system generated sentence describing the scene.

the caption captures the general idea of the video, but it is missing some key points. In this specific dataset, the problem is divided into first segmenting the video and after doing the segmentation captioning each shot. In the figure, the sentences represent the most probable output for each example shot. It can be observed that the model does not include the exact data that the game is lacrosse. Also, there is confusion about the fact that one of the speakers is holding a ball. Overall, in general, the sentences aren't human-like; however, they provide a reasonable initial description. To narrow our search to the task at hand, we carried out our study using the TRECVid dataset as it only focused specifically on the task of providing a video description of the whole short video and as the systems are evaluated using different metrics, one of them being a human evaluation of the caption, compared to the other challenges. This chapter describes the approach we developed for the TRECVid video to text task 2019, specifically the free-text generation sub-task. This sub-task consists of generating a textual description using only the information extracted from the videos. This exercise aimed to develop and test a video captioning model and try it in a real challenge and then analyse the results to find which are the most influential ways to improve the models. This is related to the research question *RQ1: What changes can be made to a video captioning model to improve it in a general way?*.

We tackle the problem using a commonly applied BLSTM network with an alternate enhancement mechanism. To improve the model, we study the effect of using different datasets and features. One of the main problems of the video captioning challenge is the vocabulary size, which adds another level of complexity, as the model needs to produce a rich vocabulary without previous knowledge of the scene. Therefore, we also discuss using an image captioning module to guide the initial text obtained from the video. The closing part of this chapter presents the areas identified as possible candidates for improvement.

Section 3.2 introduces the video captioning models and provides a concise literature review. Section 3.3 describes the basic model and the changes introduced.

Section 3.4 presents the datasets used. Section 3.5 concludes with the results and the policy used to train the models.

3.2 Video Captioning

We decided to explore the encoder-decoder strategy for TRECVid-VTT as it is the most common one for text generation in the literature. The model used is based on [Peris et al., 2016], which uses LSTMs and an attention model to encode the frames and decode the textual description. The model was selected because the code is available online, making it easy to implement and extend. It also is designed around a general architecture and implemented in a framework which makes it possible to try a variety of techniques. In terms of performance, it obtains close to state-of-the-art results. From there, we explore the use of different techniques to improve the baseline results. Examples of these techniques are to replace the BLSTM with the popular Transformer model or try to increase the amount of training data with additional datasets. Our main novel idea is to include text as input to provide a base description of the model.

3.3 Model

This section presents the baseline model and the improvements that we explore. As mentioned above we use the work of [Peris et al., 2016] as a baseline. A visual representation is show in Figure 3.2.

The model consists of a BLSTM for encoding the input features. This is one of the most common architectures used in the task of video captioning which then helps identify typical problems in most models. We use the C3D [Tran et al., 2014] as input to this BLSTM which is well known for capturing the temporal information of a video. C3D is a convolutional network with 3 x 3 x 3 convolutional kernels, train in video activity recognition datasets. This deep network is the focus in extracting spatial and temporal features, to encapsulate motion, actions, and

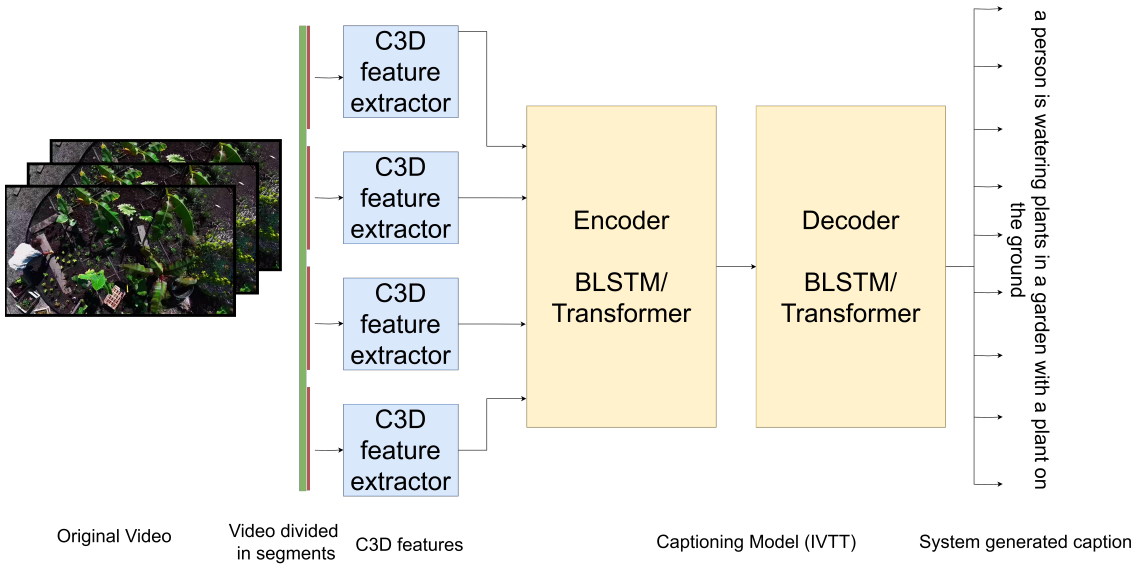


Figure 3.2: Schematic representation of the model. The video is divided into segments; these segments are converted into feature vectors using the C3D model. The final vectors are the input to the encoder-decoder architecture which is in charge of predicting the final caption.

human-object/human-scene relationships. This is followed by soft attention, which is fed again to a final LSTM. The sentences are then produced from a softmax function on top of the last LSTM. A beam-search [Sutskever et al., 2014] method is used to find the sentences with the highest probability.

A variation we explore involves predicting a caption from the middle frame and using it as an additional input for the model. The intuition behind this idea is to obtain an initial sentence from which the model can generate a more complex one.

We use the model from [Vinyals et al., 2016b] to generate a caption, which is then embedded into word spaces. This goes through an LSTM to encode the sentences and it is appended to the input vector of the soft-attention model. One noteworthy aspect is that we share the same embedding between the input and the output words to help the model identify the same words.

Let h_t^{text} be the output of the LSTM that encodes the text input, such as:

$$h_t^{\text{word}} = f(h_{t-1}^{\text{word}}, x_{t-1}), \text{for} \quad t = 1...N, \tag{3.1}$$

where N is the length of the sentences, x_t is the t word of the sentences embedded

in the Glove's spaces. Then we can define all the sentences as:

$$\mathbf{h}^{\text{sentence}} = [h_1^{\text{word}}, h_2^{\text{word}}, \dots, h_N^{\text{word}}]^\top. \tag{3.2}$$

Then the new attention becomes:

$$\alpha_i^j = \mathbf{v}^T \tanh\left(\mathbf{W}_1 \mathbf{h}^{att,j} + \mathbf{W}_2(\mathbf{z}_i \parallel \mathbf{h}^{\text{sentence}})\right), \tag{3.3}$$

where $\mathbf{v} \in \mathbb{R}^V$, $\mathbf{W}_1 \in \mathbb{R}^{V \times N}$ and $\mathbf{W}_2 \in \mathbb{R}^{V \times T}$ are learnable parameters of the model and tanh function operates element-wise. $\mathbf{z}_i$ is the i vector by the image encoder. Let $h_i^{att,j}$, $j = 1, \dots, M$, denote the i-th $LSTM^{att}$ output component at time t and $\mathbf{h}_t^{att,j} = [h_1^{att,j}, h_2^{att,j}, \dots, h_N^{att,j}]^\top \in \mathbb{R}^N$.

Attention mask coefficients β_i^j are computed as a softmax function of α_j^i, namely,

$$\beta_i^j = \frac{\exp(\alpha_i^j)}{\sum_{j=1}^M \exp(\alpha_i^j)}, \qquad \sum_{j=1}^M \beta_i^j = 1. \tag{3.4}$$

The attention vector $\mathbf{o}_i \in \mathbb{R}^M$ is computed as a weighted sum of the $LSTM^{att}$ output vectors $\mathbf{h}_i^{att} = [h_i^{att,1}, h_i^{att,2}, \dots, h_i^{att,M}]^\top \in \mathbb{R}^M$, that is,

$$\mathbf{o}_i = f_{att}\left(\mathcal{H}^{att}, \mathbf{z}_i \parallel \mathbf{h}^{\text{sentence}}\right) = \sum_{j=1}^M \beta_i^j \mathbf{h}_t^{att}, \tag{3.5}$$

where $\mathcal{H}^{att} = \{\mathbf{h}_i^{att}\}_{i=1}^N$.

We investigate improving this technique by replacing the LSTM in the encode-decode architecture with two Transformers [Vaswani et al., 2017], one for the encoding step and another for the decoding step. These are well-known for improving results and are better at capturing the temporal dependency between frames, however, they are also very demanding in terms of the resources needed for training.

3.4 Dataset

The main dataset used in our experiments is the TRECVid-VTT dataset. We use the annotations from the previous years (2016, 2017 and 2018) and the videos from 2019 without the sentences. In the three previous years, the datasets follow the same pattern: the videos come from Vine and have a length of no more than 10 seconds. Each year the organisers annotated around 2000 videos. In 2019, the only difference was that the videos come from Vine and Flickr but the other characteristics are similar in terms of length, quality or topic of the videos. There are five or fewer annotations per video and they try to reflect the *"who, what, where and when"* as depicted in the video in a short sentence.

The second source of data is TGIF [Li et al., 2016] which consists of 100k animated GIFs with 120k sentences describing their content. The GIFs come from Tumblr from May to June of 2015. On average each GIF lasts 3.65 seconds and contains 40.62 frames. One sentence per clip is provided in the training and validation sets and three for the test set. This dataset is highly compatible to TRECVid in terms of both the content and format of the videos.

MSVD and MSR-VTT are both datasets provided by Microsoft. MSVD contains around 2000 videos and MSR-VTT have 10000 videos. Both of them have around 20 English captions per clip and each clip is 10 to 15 seconds long.

The last dataset used is VATEX which consists of 41,300 videos with about 10 reference captions per video. Similar to the previous datasets the clips are around 10 seconds long.

3.5 Experimental Results

This section presents an analysis of the results of the different variations of our approach.

3.5.1 Experiment Setup

The model with the new input is going to be referred as Image-Video-To-Text IVTT-BLSTM in the following sections. These denotes the base architecture described previously using BLSTM for both the encoder and decoder.

We report performance using three of the widely used metrics, BLEU [Papineni et al., 2002], CIDEr [Vedantam et al., 2015] and METEOR [Banerjee and Lavie, 2005], described in Chapter 2. BLEU computes the geometric mean of the n-gram matching word count between the references and the candidate sentence. In our case, we report the BLEU-4, which refers to applying BLEU with 1-gram to 4-gram. The main aim of CIDEr is to capture the consensus between the annotations and the reference sentence. METEOR computes the mean of weighted unigrams' precision and recall. It also includes functions to evaluate stemming and synonyms.

We train using stochastic gradient descent (SGD) with ADAM optimizer [Kingma and Ba, 2014], and use beam search of 6. The C3D model that we used to extract the deep features for the input was pretrained in the Sports 1M dataset.

During the first phase where we train the models and compare them, we split the datasets in the following way: TRECVid-VTT 2016 and a 50% of TRECVid-VTT 2017 are used for training, then the rest of TRECVid-VTT 2017 is used for validation and finally TRECVid-VTT 2018 is the test set. When using TGIF we use the whole dataset for training.

Table 3.1: Base model and transformer model trained without the TGIF dataset.

	BLEU-4	CIDEr	METEOR
IVTT-BLSTM	0.0014	0.0730	0.1257
IVTT-Transform	0.0010	0.0660	0.1160

3.5.2 Replacing the Base LSTM with a Transformer

For the first round of experiments, we don't use the TGIF dataset only the data from TRECVid. However, we have observed that the models tend to overfit to the

training data early on. Further experiments have shown that one of the main causes of this issue is the insufficient amount of data. In table 3.1 we can see the results for the baseline model and the variation with the Transformer. We replace both encoder and decoder with Transformer models. As described in 2.3 they are popular models to solve problems involving long sequences as using the self-attention module they can make use of the whole sequence for each output token. We expect that it will give a more in-depth context when selecting each word in the final output sentence. IVTT-Transform is an expensive model to train and the results were not as good as expected at first. From other works, it is known that Transformers suffer in scenarios which don't have enough samples as in this case. As such, we decided to discard further experiments with this model and focused on other variations of the baseline. In future work, it will be interesting to make use of the Vision Transformer to use pretrain model in image/video data.

Table 3.2: Base model (IVTT-BLSTM), image caption model(Image captioning) and base model with image caption text as input (IVTT-BLSTM-Text).

	BLEU-4	CIDEr	Meteor
Image captioning	0.0025	0.0420	0.1111
IVTT-BLSTM	0.0020	0.1012	0.1333
IVTT-BLSTM-Text	0.0013	0.1090	0.1358

3.5.3 Data Augmentation with Additional Datasets

In this experiment, we include the TGIF dataset to increase the amount of data. We also explored the addition of the captions extracted from the Image Captioning module into the model as input. The results can be found in 3.2. The first thing to notice is that, as expected, more data improves the baseline. Apart from this we also can see that the base Image captioning model performs well even if we do not retrain it to adapt to the videos and sentences of the dataset. However, if we use these sentences in our model the performance decreases in our metric of references, BLEU-4, and only improves slightly although adding quite a few extra steps and

increasing the size of the general model. As the improvement isn't sufficient to be noticed across all metrics we decided to remove it from the final solution.

Table 3.3: Final results of the model submitted to TRECVid 2019.

	BLEU-4	CIDEr	METEOR
IVTT-BLSTM	0.0045	0.0350	0.1415

3.5.4 Competition Submitted Set-up

We only submitted our baseline run to TRECVid 2019 [Awad et al., 2019]. This model was trained on the whole TRECVid-VTT 2016 and TRECVid-VTT 2018 plus 50% of TRECVid-VTT 2017. For validation, we use the rest of the TRECVid-VTT 2017 as in the other experiments and predict all the test datasets. Another noteworthy aspect of our work is that we use Glove embedding [Pennington et al., 2014] for the output words. The results can be seen in table 3.3.

3.5.5 Robustness Study: Shuffle Frames

To test the robustness of the model, we shuffle the input frames. The idea behind this experiment is to check if the temporal information of the video is indeed relevant to this model. By shuffling the frames the model is going to process the video in another order so if the model is robust to this change means that it only use the image data. Table 3.4 shows these results. In a small dataset like the 2000 videos of TRECVid, shuffling does not have a significant effect. However, when the number of videos and captions is increased, there is a significant drop in all the scores, ROUGE-L being the least affected. These prove that when the model is well trained is actually using that temporal information.

3.5.6 Using Additional Datasets

This experiment explores using more data and observing if there is an improvement. In these experiments, MSVD [Chen and Dolan, 2011], MSR-VTT [Xu et al., 2016]

Table 3.4: Results on the TRECVid 2018 dataset when shuffling the input frames.

	BLEU-4	CIDEr	METEOR	ROUGE-L
IVTT	0.009	0.019	0.059	0.158
IVTT-Shuffle	0.010	0.011	0.059	0.166
IVTT-TGIF	0.071	0.174	0.123	0.288
IVTT-TGIF-Shuffle	0.015	0.035	0.069	0.188

and VATEX [Shi et al., 2022b] datasets are added to the data, and a study of their effect on the metrics is performed. Table 3.5 shows the results. In terms of a single dataset, TGIF dataset is the one that gets the best performance. Two of the biggest datasets produce significant improvements compared to the other. However, Vatex doesn't add any improvement to the base model. This can be understood by having the biggest corpus with the richest captions, covering 600 human activities.

Table 3.5: Results on the TRECVid 2018 dataset adding different datasets.

	BLEU-4	CIDEr	METEOR	ROUGE-L
IVTTT	0.009	0.019	0.059	0.158
IVTT-TGIF	0.071	0.174	0.123	0.288
IVTT-MSR-VTT	0.057	0.152	0.150	0.264
IVTT-MSVD	0.011	0.033	0.070	0.190
IVTT-VATEX	0.009	0.025	0.064	0.176

3.5.7 Metric Study

Table 3.6: Correlation between metrics on the TRECVid 2019 [Awad et al., 2019].

	CIDEr	CIDEr-D	METEOR	BLEU	STS	DA
CIDEr	1.000	0.972	0.963	0.902	0.937	0.874
CIDEr-D	0.972	1.000	0.967	0.969	0.852	0.832
METEOR	0.963	0.967	1.000	0.936	0.863	0.763
BLEU	0.902	0.969	0.936	1.000	0.750	0.711
STS	0.937	0.852	0.863	0.750	1.000	0.812
DA	0.874	0.832	0.763	0.711	0.812	1.000

Following the TRECVid 2019 benchmarking campaign, the organisers reported the correlation of the metrics against each other at the system level for the subtask of text generation – see Table 3.6. STS refers to Semantic textual similarity and

DA makes refers to the human judgements, also know as Direct Assessment. It can be observed that most of the metrics used are highly correlated. The one with the lowest correlation is BLEU which, as explained previously, is the most simple. However, it can be seen that all the metrics result in low correlation with human judgment (DA) in this setup. This leads to the conclusion that none of them capture the details humans use to evaluate the captions. The metrics that obtain the closest correlation are the ones based on CIDEr and the text-similarity one. CIDEr is the only one of the set of metrics that evaluate the consensus between the human references, which means that the differences between the human annotations greatly influence the final evaluation.

3.6 Summary

In this chapter, we proposed a model to generate captions from short videos. We trained in a diverse datatset with video from the web. We explored the use of image captioning models to establish a base sentence from the model. We also tried to replace the LSTM in the architecture with a Transformer.

The change to a Transformer-based architecture does not provide an improvement in our experiments. This is probably because we had a limited number of samples and couldn't conduct many experiments due to the time-consuming nature of working with this model. The sentences from the image caption module obtained the best performance compared to our model but they are limited by the pretrained vocabulary. Adding them to our model does not yield results as good as we expected. This is likely because the approach we follow to include them is quite simple. Concatenating the whole sentences to the image input vector adds too much redundancy. It also includes some level of hallucination as it can describe objects that may not be in the scene at the relevant time.

None of the variations we explored resulted in an improvement in performance with respect to the metrics, which is why we decided to only submit the baseline with a new training methodology. However, there are avenues to explore to improve

performance. We observe that the model shows improvement only when certain types of data are used. One difference we observe comparing our model with the winning team's models is the metric used as a reference to perform the final model selection to submit to the workshop. We use BLEU-4, which was one of the main ones used in the literature, while most of the high-scoring teams use CIDEr or CIDEr-D as their main reference metric.

3.6.1 Better Evaluation as the First Step to Improve Video Captioning Models

A significant number of possible improvements were identified by working on this challenge. One of the main weaknesses of the video captioning model is the diversity in the construction of sentences. Models tend to form simple sentences, starting with something like "A man ...". It is also difficult to see if a model is learning well during training. Most of the time, the models are trained to (ideally) reproduce the same sentences as human annotators.

There are multiple paths to explore to try to solve these problems. The text generator could be improved by generating more sentences, choosing a single one as a post-processing step, or generating more reference captions with paraphrasing and training in the augmented set. We could also explore additional features and additional multipurpose datasets. For example, the Visual Genome dataset provides multiple objects and relationships in the image domain, which could be included in a model as clues for the final sentences. Multi-objective or Multi-task learning could provide this information without using complex datasets; we could use only a dataset for each task or objective.

However, none of these approaches gives a straightforward way of interpreting whether the results are improving. We would still need to rely upon the traditional metrics, which in most cases, do not take into account the information in the video. This is the basis of our motivation for exploring whether the creation of a new metric could lead to improving the model. The ideal metric should follow a similar

evaluation process as with a human. It needs to be disconnected from looking into how similar the candidate caption is to the reference sentences. It also should include a set of critical concepts that should be included in the final results and, after detecting these keywords it should then evaluate the level of detail required for that concept. For instance, if there is a cat in the scene it should be included in the caption; then the metric should reflect how in-depth the description of the cat is and what it is doing deepening the relevance of the cat. A better caption can then be used to improve the evaluation of the models at test time and improve the training.

A better judgement of the quality of the models could help to diversify the vocabulary as it will be focused on the general idea being described instead of the structure and the specific words used by the model. A metric such as this could help identify when the model fails to capture the key concepts or needs more details.

Of course, the ideal scenario would be to have a human in the loop but this is unrealistic in practice as it is both time-consuming and expensive. As such the best metric is one that provides results most similar to human judgements. The main focus of the following chapters is how to improve the evaluation process and how to imitate human decisions.

Chapter 4

BERTHA: Video Captioning Evaluation via Transfer-Learned Human Assessment

4.1 Introduction

As discussed in the initial description of the metrics and the conclusion of the previous chapter, evaluating video captioning systems is a challenging task as there are multiple factors to consider, like the fluency[1] of the caption or the human interpretation or bias of what is relevant. As explained, the metrics typically used for video captioning are borrowed from other tasks like Machine Translation or Image Captioning. Metrics like BLEU [Papineni et al., 2002], METEOR [Banerjee and Lavie, 2005] or CIDEr [Vedantam et al., 2015] were designed for Machine Translation. These metrics are normally based on computing the overlap between the caption and the reference sentence. They fail to correlate with human judgement as they do not consider the context. New metrics like SPICE [Anderson et al., 2016] try to address this problem. However, these have their own problems and limitations as they only reflect what a small number of annotators think is a good caption. Validating the accuracy of a given metric is carried out by correlating it with the

[1] Fluency represents the language proficiency use in the sentences.

Human Caption: a man in a white helmet and blue shirt is handed an orange rope and tosses the end of it over a steep descent through a flowing stream

System caption: A man is climbing down a rock wall with a rope and a woman

CIDEr: 20.5 Human assessment: 1.0

Human Caption: A man in a black cap, bright Hawaiian shirt, and jeans sits on a skateboard with his hands straight up and rides it down a concrete grade till he stops in the grass

System caption: a person wearing a yellow shirt and a hat is riding a snowboard on a mountain

CIDEr: 59.9 Human assessment: 4.7

Figure 4.1: An automatic metric (CIDEr in this case) fails to evaluate the caption as a human would. Note that the human assessment scores are both low as the range is between 0 and 100.

human assessment of the quality of system-generated captions. In other words, by examining the degree to which metric scores for captions correspond with the human opinion of the same caption. In this chapter, we present a new automatic metric that directly learns from human evaluations to maximise the correlation with human judgement.

This chapter presents a new evaluation method based on a deep learning model. Our new method of evaluating video captioning systems makes use of pre-trained language models to learn the aforementioned correlation. In general terms, language models aim to predict the next word or character within a text. Recent work has seen substantial advances by models trained on vast quantities of data on language processing tasks and the emergence of well-known language models such as BERT [Devlin et al., 2018] and GPT-3 [Brown et al., 2020b].

By fine-tuning a pre-trained language model, it is possible to obtain excellent results in a huge variety of tasks [Devlin et al., 2018], which overall is less computationally demanding than training a model from scratch for a given problem. BERT has been shown to work well for tasks similar to video captioning such as

MT evaluation [Sellam et al., 2020], and thus we adopt this model in our work.

Since existing metrics do not correspond well with human judgement about what may or may not be a good quality caption 4.1, our proposal is a new metric based on BERT and fine-tuned with an accurate human evaluation of video captions. We test the performance of our metric on data from past TRECVid [Smeaton et al., 2006] video-to-text challenge tasks. As human judgement is sometimes unreliable, multiple human assessments are collected for captions to reduce any uncertainty contained within our training and test data.

The specific contributions presented in this chapter are the following:

- A new video captioning metric trained on human judgements. It uses a pre-trained language model to ease the training phase.

- We evaluate its performance in a challenging dataset against the commonly used metrics and show that the new metric is consistently in the top-performing metrics.

- We study different evaluation scenarios and test the limits of the metric to understand its behaviour.

In Section 4.1 the general idea behind this work is introduced. In Section 4.2 the related work in the literature on metrics, language models and learned metrics is presented in detail. Section 4.3 describes the general problem addressed in this chapter with an explanation of BERT and the modifications made to produce BERTHA. Section 4.4 presents the evaluation procedure with a description of the dataset used and the data collection process along with the results obtained. Finally, Section 4.5 presents a conclusion for all this work.

4.2 Related Work

4.2.1 Video Captioning Metrics

In simple terms, the evaluation of captions can be divided into two factors: the quality of the generated text and how well it describes the scene. For the first, there are many metrics in the NLP domain that can evaluate this to a certain extent. For instance, language models like BERT [Devlin et al., 2018] or GPT [Radford et al., 2018], can measure the probability that a set of words can occur in a sentence. Also, part-of-speech-tagging or parsing [Manning et al., 2014] can be used to find the category of the word, e.g. noun, verb, etc., and identify mistakes. Finally, sentiment analysis can evaluate sentences more subjectively and identify which sentiment the phrase is transmitting to the reader [Devlin et al., 2018]. However, the second factor is substantially more challenging as is more subjective to the human reading the caption. We can define concepts that are used to evaluate the caption but are difficult to quantify like for instance the level of detail needed by a caption to be considered "complete".

Automatic evaluation techniques use a variety of approaches to measure the generated captions, and usually, people refer to a set of metrics as they can complement each other. In most metrics, the main points to consider are fluency, accuracy in describing the content of the video and similarity to the human references. Some widely used metrics in the literature are:

BLEU [Papineni et al., 2002], which was initially developed for machine translation, is a metric based on computing the geometric mean of n-grams match counts. ROUGE [Lin, 2004] is similar to BLEU, which also uses n-grams. The difference is that ROUGE considers the n-gram occurrences in the total sum of the number of reference sentences while BLEU considers the matches in the sum of candidates. All these metrics only look for exact tokens which is a big disadvantage when rephrasing. In some cases, we do not have only a single reference per video but a set of them. A metric like METEOR [Banerjee and Lavie, 2005] performs better in these

situations. METEOR compares exact token matches, stemmed tokens, paraphrase matches and semantically similar matches using WordNet synonyms. Another aspect to consider is the difference between each of the available references. Metrics like CIDEr [Vedantam et al., 2015] try to take this into account. CIDEr measures the consensus between system-generated captioning and the set of reference sentences. Another approach is SPICE [Anderson et al., 2016] which is a metric that uses a scene graph to describe the semantics allowing it to identify similar sentences in a more generic way than the other metrics.

These will be the metrics that we are going to use to compare our method against them.

4.2.2 Human Judgement

Scores produced by the metrics described above aim to evaluate the similarity between a system-generated caption and a human reference caption. However, they cannot detect if they are describing the same scene. For instance, there can be multiple details in a video clip that are not important for the human annotator, like the colour of an animal, but the model can decide to include it in the final caption. An example of this problem can be seen in figure 4.1. In these two pictures, an example is shown of captions where a popular automatic metric (CIDEr) fails to evaluate the caption as a human would. In the first case, the system caption doesn't contain as much detail as the human caption, and it includes a woman who is not present in the scene. In the second case, the system caption seems like it does not describe the same scene; however, it contains some keywords in the human caption.

Most metrics also have problems in terms of robustness, systems being penalised for legitimate differences in word order, using word replacement or change in the given score when using a shorter version of the original sentence. As it quickly becomes difficult to compare automatically, human judgement is normally involved in comparing the metrics. This requires direct human evaluation of the caption quality or human-generated captions used as a reference. When a human-generated

reference caption is employed, a quantitative evaluation of sorts can be carried out that measures how similar the system-generated caption is to the human reference. Then metrics can be compared based on how well they correlate to this human judgement.

These metrics are used to define the loss function that are used to train the models. This produces a mismatch when trying to include one of these losses as none are differential, an essential condition for the back-propagation step and gradient descent as explained in Chapter 2. They are usually only used as an evaluation metric and a different loss is used to train the model. Most of these losses try to include some information that will guide the model to get closer to the evaluation metric, like trying to predict the SVO [Zheng et al., 2020a] or finding the objects in the scene [Zhang et al., 2020]. Another solution to this problem is to use reinforcement learning (RL). In this kind of approach, a metric uses a reward function which will only be evaluated at specific points, and the policy gradient directs the model, removing the need for these metrics to be differential [Zheng et al., 2020a]. In these cases, typically, CIDEr is used as the base metric or any of its variations. However, it can cause the model to overfit this metric. This scenario is bad because we can't distinguish how well it will perform in other datasets, and we can't analyse the results. These datasets aren't general enough, so we normally want to be able to add some kind of explainability of why it failed and applied it with other complementary data/metrics/loss. Finally, some methods do not focus on the evaluation metric only; they aim to produce human sentences using NLP techniques more generally. They make use of external models like pre-trained language models to generate sentences imitating the reference captions [Zhang et al., 2020]. This approach can be extended so that models can use pre-detected objects in the scene to guide the caption. This proves to work well in the scenarios investigated. Still, it adds complexity overall as now part of the model is dedicated to solving other computer vision or NLP tasks like object detection or SVO identification. Moreover, these do not help compare the generated captions as they may use different words

or sentences to describe the same scene as the structure is more limited than in a free text scenario like captioning.

There have been some studies on training models in the machine translation domain to imitate human evaluations of sentences with actual human scores from annotators. In this case, most methods aimed to solve this problem with hand-crafted features and classical machine learning methods. However, nowadays, most methods used in the WMT Metrics Shared task [Mathur et al., 2020], a task in the Conferences on Machine Translation to compare metrics, have started to include some form of deep learning, specifically BERT as a base architecture [Sellam et al., 2020, Shimanaka et al., 2019] and obtain good results. Also, some initial studies have been done on using BERT in image captioning [Zhang* et al., 2020]. In this case, good results are obtained compared to the standard metrics but lose out against a trained metric that tries to predict if a caption comes from a human.

Similar to the machine translation domain, some studies are looking for ways to learn a metric from human judgements. Datasets like Microsoft COCO [Chen et al., 2015], or PASCAL [Vedantam et al., 2015] have some captions annotated and judged by humans. In the case of COCO, the captions are evaluated with the following criteria: percentage of captions that are better or equal to humans; percentage of captions that pass the Turing Test; average correctness of the captions; the average amount of details; percentage of captions that are similar to human references. All the other datasets only use scoring from 1 to 5 on how similar a caption is to a human reference or how well it describes the clip. Even the more complex evaluation carried out in COCO falls short of establishing a satisfactory ranking between all the captions as the range is short. The study of trained methods in these datasets with human judgement has proved to correlate better than the image captioning metrics [Yi et al., 2020, Cui et al., 2018].

4.2.3 Language Models and Transfer Learning

Language models are commonly employed in a wide range of NLP tasks and hugely influence text fluency produced by automated systems. They are models which predict the probability that a sequence of words can occur in a sentence. Most architectures involve training on huge unlabelled corpora such as Wikipedia and news articles, literary text and language-based web data [Brown et al., 2020b]. This type of large architecture aims to decouple the model from the task so that the same model can be applied to distinct tasks.

One example of this is BERT [Devlin et al., 2018] which is a bidirectional language model based on Transformers [Vaswani et al., 2017]. BERT has become very popular thanks to performing well in a diverse set of tasks like semantic textual similarity or sentiment analysis. To do so, it uses a technique called transfer learning. These techniques consist of using an already trained model to solve another task hoping to use the already learned knowledge in the new task. In BERT, the base model is trained on unlabelled textual data using two techniques: masked language modelling and next sentence prediction. Then this pre-trained model can be retrained for multiple tasks. As mentioned above, this model has proved to work well in multiple similar tasks like machine translation evaluation [Sellam et al., 2020, Shimanaka et al., 2019] or image captioning evaluation [Cui et al., 2018]. For this reason, we use it as the base model for our experiments.

4.3 Evaluation of Video Captioning via Transfer Learning

When automating the evaluation of tasks aimed at human users, such as video captioning, a common approach is to compare system outputs with one or more human-generated reference(s). Our proposed model only uses a single pair of caption and human references each time. This simple scenario is helpful as it can work with multiple human-generated references or with only a limited number.

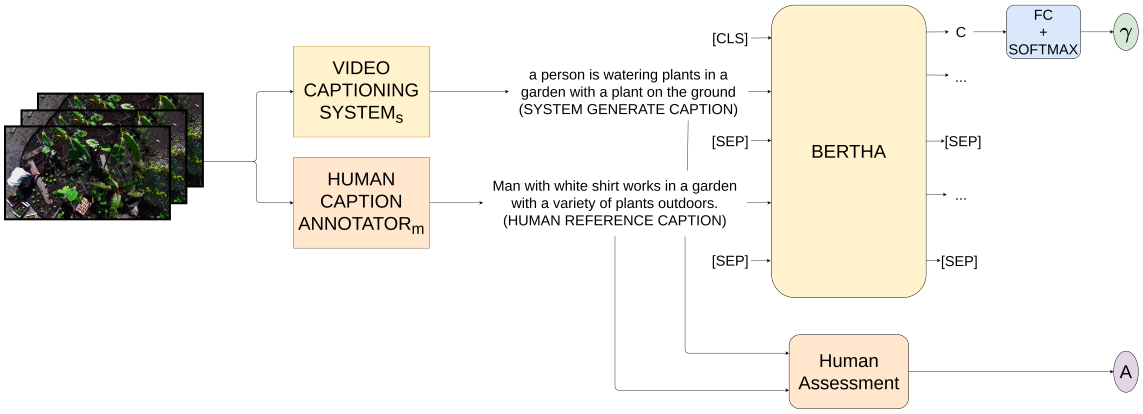


Figure 4.2: Evaluation with the proposed BERTHA video captioning evaluation method.

Given a caption output by a system and a human-generated reference, our proposed evaluation model aims to learn the distance in meaning between the two strings of text. In order to train the model, we employ captions output by systems participating in past benchmark shared tasks and carry out a human evaluation of the quality of the captions. The details of the data collection process are explained in the next section.

Human evaluation takes the form of a crowd-sourced worker watching the video, reading the caption and rating how well the caption describes what took place in the video. Thereafter, a set of system-generated captions are assessed by humans who watch the video in question and provide a rating that reflects the quality of the system-generated caption. In these annotations, a score was given depending on how similar the description was to a reference sentence. The system caption, a corresponding human-generated reference caption (also taken from data employed in benchmark tasks) and the human assessment score we collect are then employed to train the model, where the aim of the model is to maximise the correlation between the learned metric and the human ratings of caption quality.

We define $V = \{v_1, ..., v_N\}$ as the set of $N \in \mathbb{N}$ videos. Notice that we can have $M \in \mathbb{N}$ different reference captions for a single video. We employ $S \in \mathbb{N}$ captioning systems to produce captions for videos and carry out $\mathrm{A}_{N',M',S} \in \{x \in \mathbb{R} | 0 \leq x \leq 1\}$ human evaluations of video captioning systems where $N' \leq N$ and $M' \leq M$. It

should be noted that not all the videos in the source dataset are annotated. As mentioned previously, the goal of training the model is to maximise the correlation between predicted scores ($\Gamma \in \mathbb{R}^{N',M',S}$) and the given human qualitative scores:

$$\underset{\theta}{\text{argmax}}\ \mathrm{r}(\Gamma, \mathrm{A}) \tag{4.1}$$

Even if we consider all the data of the task used, it is still a relatively small dataset. As proved in the previous chapter, the data available is not enough to train a model to accurately produce the captions so it can also be argued that it is not enough data to train the evaluation process from scratch. This is why we decide to use a pre-trained language model and fine-tune it to our data. Figure 4.2 provides a depiction of the process involved in training our proposed video captioning evaluation metric, BERTHA compared to the data collection process. The input to the model is a system-generated caption and a human reference caption. The Video Captioning System and Human Caption Annotator correspond to one of the systems M in the dataset and one of the human references used S, with $m \in M$ and $s \in S$. The bottom part of the figure refers to the human assessment used to obtain the human score A. Two symbols are added to the captions: one to mark the beginning of the sentences [CLS] and another to mark the end of the individual phrase [SEP]. The first output of the model, denoted by the symbol C, is fed to a Fully Connected layer (FC) with Softmax activation to obtain the predicted score γ.

The base model used is BERT (Bidirectional Encoder Representations from Transformers) [Devlin et al., 2018]. The core module of BERT is a set of Transformer blocks [Vaswani et al., 2017] which, in one of its configurations, can have two sentences as input and outputs two sets of symbols corresponding to these input sentences. The two input sentences (system and human-generated captions) Γ and A are tokenized using WordPieces [Wu et al., 2016]. WordPieces is an algorithm to represent groups of subwords as tokens, the vocabulary starts small and is extended iterative with the most frequent combination of symbols in the same collection of documents.

From the model's output, only the first symbol C is used, and all the rest is discarded. A multilayer preceptor with a single output is attached to the C symbol to obtain the final score. C is defined as $C \in \mathbb{R}^H$ where H is the hidden side of the Transformer blocks [Vaswani et al., 2017]. We add a layer on top, with $W \in \mathbb{R}^H$ and $b \in \mathbb{R}$ as weights and bias, to compute the final score γ:

$$\gamma_{i,j,t} = \text{softmax}(WC_{i,j,t} + b), \tag{4.2}$$

with $i \in N', j \in M', t \in S$.

BERTHA uses BERT-base, the smaller model of the two presented in the original paper. It consists of 12 layers, 768 hidden states and 12 heads. BERT-base is trained with the configuration described in [Devlin et al., 2018] using masked language modelling loss and subsequent sequence prediction loss. Using this model as a baseline, we fine-tune it to our task using a regression loss (Mean-Squared loss). BERTHA is fine-tuned end-to-end.

4.4 Experimental Results

4.4.1 Dataset

To evaluate our approach, we use data sets from past TRECVid [Awad et al., 2020] benchmark video to text tasks (VTT) from 2016 [Smeaton et al., 2006],2017 [Awad et al., 2017],2018 [Awad et al., 2018],2019 [Awad et al., 2020] and 2020 [Awad et al., 2021]. Approximately 2,000 videos are available each year from Vine and also from Flickr and V3C2 in later years. The videos cover multiple topics and multiple camera angles. The videos are short clips about 6 seconds long; some videos can be up to 10 seconds long in the last year. Each of these videos has been annotated with a caption by between two and five human annotators. The original task was divided into two subtasks, to generate a caption for each video and to find and match human-generated captions with the correct video. We use the data to train

and evaluate the model taken from captions produced by systems that submitted results for the first task.

4.4.2 Human Assessment

The annotations from the human assessment were collected using Amazon Mechanical Turk.[2] The process to annotate the data is as follows. First, the annotator will look at the video and compare the system-generated caption to the content of the video. Then they will give a score between 0 to 100 on how well the sentences represent the video. They will also evaluate the fluency of the caption in English. The final score obtained from the process is a combination of the two concepts. To detect consistency in the same annotators, degraded and repeated sentences are used. When the final collection is finished, the human evaluation is converted to Z-score per human annotator, the mean and standard deviation are removed from each data point coming from the same worker. For the purposes of our experiment, we divide the annotations into two sets: single annotator (SA) and multiple annotators (MA).

The SA set is collected as in [Graham et al., 2018] where only a single human annotates each pair of captions. The worker sees the video and then compares the sentence with what was seen. To validate that the annotators are reliable, a process was used where captions coming from both automatic systems and human annotators were used together. Some of the human captions are also degraded with random parts of other human captions. Thus, a poor quality worker could be identified as they will overlook the captions produced by humans and not give them a high score. Equally, they will also skip the degraded captions and assign a poor score. The final metric consists of a z-score computed as the standard deviation from the mean data point regarding its standard deviation and mean score.

The second set of human evaluations, MA, is collected in a similar manner. However, instead of using a single annotator per system-references, a minimum of 15 annotations were used. The 15 annotations proved to be sufficient to obtain

[2] urlhttp://www.mturk.com

stability in the scoring [Graham et al., 2015]. The final score is the average of these annotations.

The SA dataset has 56,088 annotations (unequally) distributed across years, and the MA dataset has 7,705 annotations equally distributed across each year. Collecting MA data is more costly than collecting SA data[3]. It should be noted that the equitable distribution of captions per year should not have a significant impact as we are equally interested in both low-quality and high quality captions.

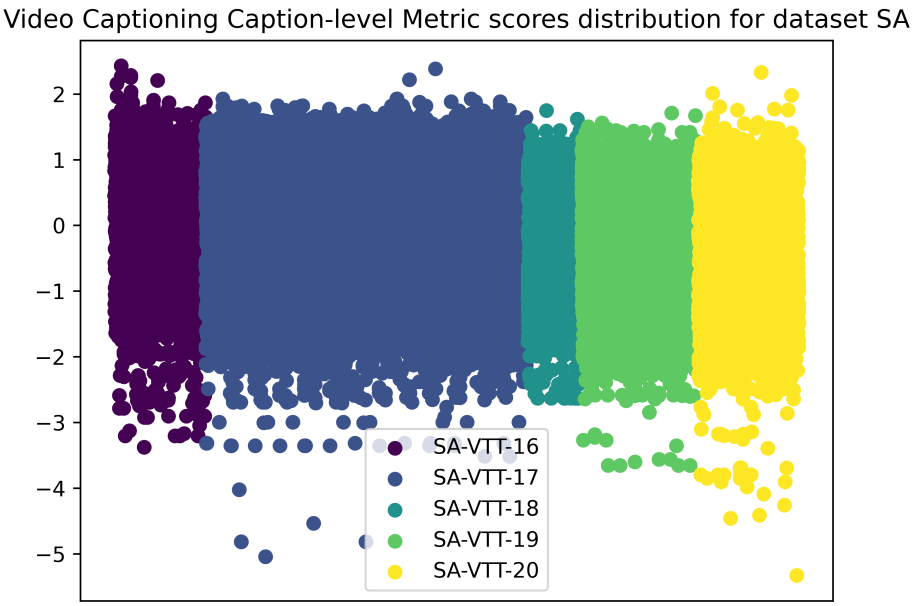
One relevant statistic is how the scores from the human assessment are distributed in each dataset. In figure 4.3, we can see the different distribution per year and dataset. SA has a mean score of -0.28, and MA has a mean score of -0.25, however, in the plot it can be seen that the deviations are very different. As both datasets include similar sentences we can deduce that system generated sentences of similar quality have a noticeable difference in the human score in each dataset.

4.4.3 Evaluation

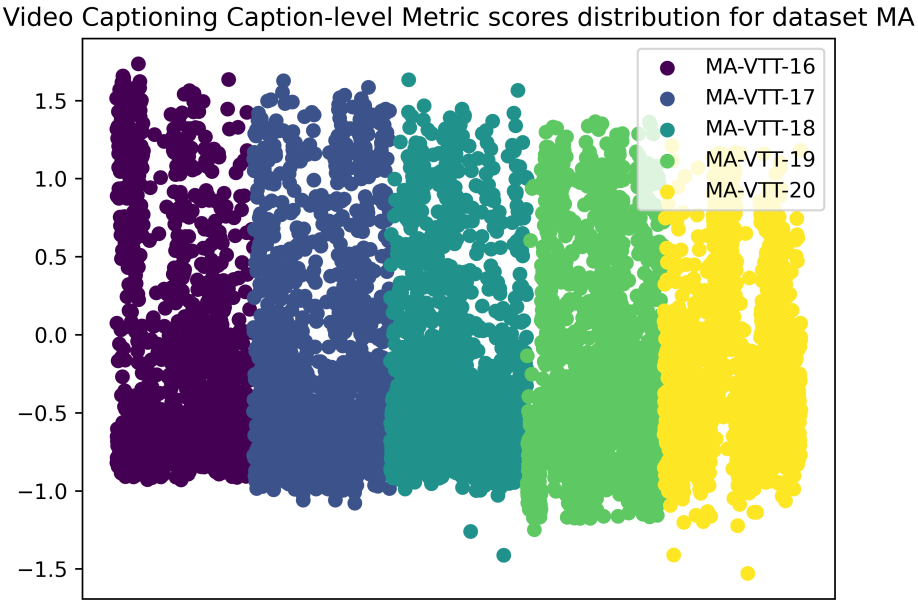
In this section, a comparison of how BERTHA works in the two datasets is presented. Evaluation is performed at the system level and the caption level. However, the method is trained only at the caption level. We focus on two primary configurations: BERTHA-SA and BERTHA-MA. Each model is trained in all the years of one of the datasets, e.g. BERTHA-SA is trained in the single annotator dataset. Each dataset is divided by each year to better represent the typical set-up of the TRECVid challenge, so we train in a set of years and save one for testing. In each table, the column represents the year used as a test. In both models, the same year used for testing is discarded from training as some system-generated sentences are shared between the two datasets.

Even if there are multiple human references, they are not standard in number. In the early years of the challenge, only two human sentences per video were used.

[3]SA has a mean of 15 tokens per human references caption, and nine tokens for the system generated captions in terms of length. MA has a mean of 14 tokens per human reference and eight tokens per system-generated caption.



(a) Score distribution of Video Captioning Caption-level Metric scores with human assessment SA for TRECVid 2016 to 2020



(b) Score distribution of Video Captioning Caption-level Metric scores with human assessment MA for TRECVid 2016 to 2020

Figure 4.3: Score distribution of Video Captioning Caption-level Metric scores with human assessment for both datasets

Also, all the videos and systems do not have the same number of human references. Because of this, the base experiment considers each sentence independently, even if they come from the same video and the same system. In further experiments, we discuss the use of multiple references.

In all the comparisons, SPICES obtained results way below what is expected. Some points may have influenced this low performance. First, we use two codes to compare and obtain the other metrics: one official from MS COCO [4] and an unofficial but highly used port to Python 3 [5] . No big differences were detected between the two and we didn't have any problem with the other metrics. However some parameters may be not configured well for SPICE. Second the evaluation in such a small dataset is normally in a small range, 0.15 - 0.20, these could influence the ordering of the captions and the difference couldn't be enough to represent well SPICE when computing correlations. Related to the small dataset, the scene graph should be quite complex in a dataset like TRECVid-VTT as there is a fairly huge vocabulary with personal names. These may add more complexity to the computation of SPICE.

4.4.4 Raw Scores vs Standardized Scores

In the TRECVid dataset, two scores can be used: the raw score as the Mechanical Turk workers annotated it and a standardised score of these, computed as the standard deviation from the mean data point regarding its standard deviation mean score. This standardized score is called the Z-score. The aim is to remove the bias added by a user when selecting the scale they use as we don't predefine what every value of the scale means. Table 4.1 shows a small test example using TRECVid SA 2019 data, with data divided as 70% train, 15% validation and 15% test. It can be observed that overall the other metrics obtain a similar correlation in both cases. However, BERTHA shows better performances while using the standardised score. For all of our experiments, we use this z-score for both BERTHA and the other

[4]https://github.com/tylin/coco-caption
[5]https://github.com/salaniz/pycocoevalcap

	Z-score	Raw Score
BERTHA	18.6	-6.5
BLEU-4	1.8	2.8
CIDEr	9.1	9.6
Meteor	9.2	10.7
ROUGE	5.8	5.9

Table 4.1: Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment for TRECVid SA 2019 participating systems; for both the raw score obtain from the annotators and the z score which is the standardize version of the raw one.

metrics.

4.4.5 System-level Evaluation

Concerning system-level evaluation, each metric takes as input the set of videos from a single past TRECVid-vtt task and one or more human-generated reference captions for that video, as well as the video captions produced by all systems participating in that task. For each participating system, the metric produces a single score corresponding to the average score of each sentence produced by this system. Taking scores produced by the metric for all systems from a single TRECVid VTT task, we calculate how well it correlates with human assessment of the same systems in terms of the SA or MA scores described above.

To evaluate the system level, we use the following formula:

$$\mathrm{r}(P, U), \forall t \in S \tag{4.3}$$

, where $P = \frac{1}{N'M'} \sum_{i,j}^{N',M'} \gamma_{i,j,t} \forall t \in S$ and $U = \frac{1}{N'M'} \sum_{i,j}^{N',M'} Ai, j, t \forall t \in S$

In table 4.2 results for the entire SA dataset are presented. The first thing to notice is that VTT-16 was the first year where this evaluation system was implemented, explaining why BERTHA and the other metrics behave differently. VTT-16 is also the earliest year, and assuming that there is an improvement in the models each year, it is also the year with the worst system-generated captions in general terms.

	VTT-16	VTT-17	VTT-18	VTT-19	VTT-20
BERTHA-SA	27.4	80.1	94.8	**92.9**	96.3
BERTHA-MA	72.0	70.6	87.8	83.7	86.3
BLEU-4	77.1	78.2	87.4	58.1	94.4
Cider	**81.1**	81.8	96.1	81.0	**97.7**
Meteor	62.8	**90.7**	**98.9**	88.7	95.8
Rouge	29.5	55.5	92.6	58.8	91.9
SPICE	-27.8	10.3	-44.7	49.8	-1.1

Table 4.2: Pearson Correlation of Video Captioning System-level Metric scores with human assessment SA for TRECVid 2016 to 2020 participating systems; metric is trained on captions from all other (non-test) years

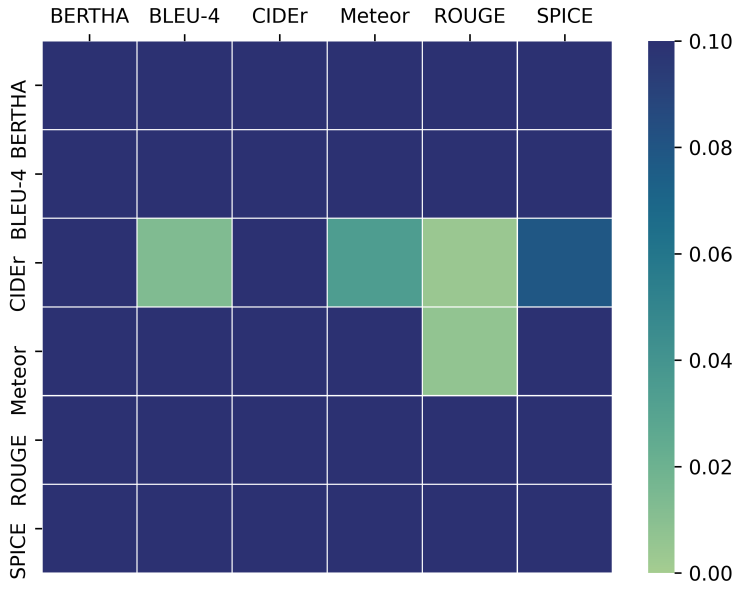
	VTT-16	VTT-17	VTT-18	VTT-19	VTT-20
BERTHA-SA	**85.7**	74.4	93.6	85.9	88.8
BERTHA-MA	74.3	71.5	88.4	88.2	86.3
BLEU-4	11.9	60.7	22.1	75.3	96.7
CIDEr	-0.1	**84.8**	93.4	**89.2**	96.7
Meteor	-44.9	79.2	94.2	88.0	**98.8**
Rouge	-65.0	05.6	**94.9**	81.1	91.4
SPICE	-35.6	-13.5	7.9	-24.0	14.6

Table 4.3: Pearson Correlation of Video Captioning System-level Metric scores with human assessment MA for TRECVid 2016 to 2020 participating systems; metric is trained on captions from all other (non-test) years

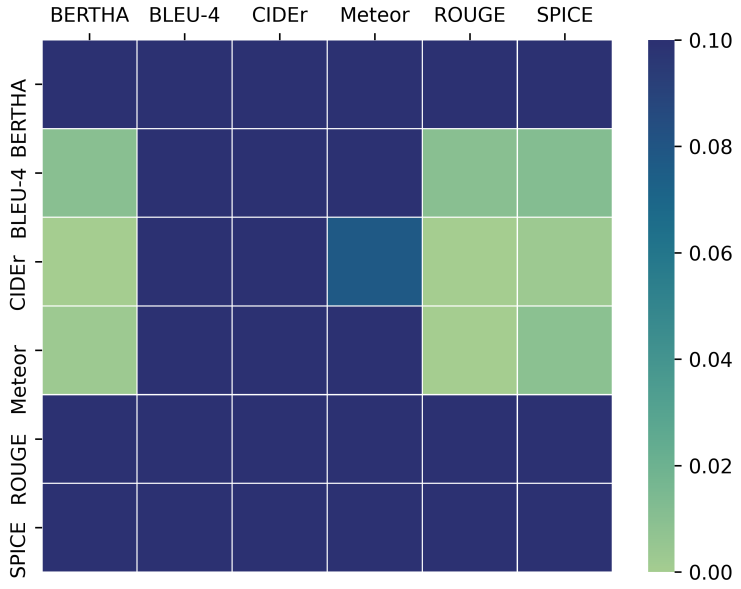
In the case of the MA dataset, which is represented in figures 4.3 VTT-16 is still the most different set in terms of results across all metrics. In this case, one of the most stable metrics is CIDEr which always gets one of the top-3 scores in all years. BERTHA gets a good score overall – it is the second or third best performing metric in most cases.

Finally, we perform a William significance test [Graham, 2015] at the system level, considering $p < 0.05$ as the references significance probability in the William test. CIDEr is the best scoring one but only obtaining a clear difference from the worst-performing ones. Differences between metrics are more pronounced in the MA dataset.

Overall the significance test proves that there is no clear winning metric, justifying using multiple metrics. CIDEr is the metric that obtains a better consistency in both datasets, but in terms of results, there are no significant differences compared



(a) William significance test of Video Captioning System-level Metric scores for the SA dataset



(b) William significance test of Video Captioning System-level Metric scores for the MA dataset

Figure 4.4: William significance test of Video Captioning System-level Metric scores with human assessment for TRECVid 2016 to 2020 participating systems. The p-value in a cell means the significance of the row vs the column in the absolute Pearson correlation.

to the other best-performing metrics.

In the case of BERTHA, we can see that VTT-16 performs worst than in the others years. From this, we can deduce that the tendency of VTT-16 is different from all the other SA years. However, the inclusion of this year as training does not seem to affect the performance of BERTHA and it obtains similar behaviour compared to the other metrics. We can argue that VTT-16 gives more examples of wrong captions, and the model aims to distinguish between good captions and incorrect captions. In the last two years, we can see an improvement in the overall scores meaning that the captions were easier to distinguish between good and bad sentences.

4.4.6 Caption-level Evaluation

For caption-level evaluation, the system is removed from the equation, and the metric is evaluated at the caption level so that for each sentence produced by each system, we obtain a score. Taking all the scores for a year from a single TRECVid VTT task, we calculate how well this correlates with human assessment of the same year.

To perform the caption level evaluation the $r(\Gamma_{N',S}, A_{N',S})$ is used in the single references per caption scenario. Some videos can have multiple references captions. In some traditional metrics, like CIDEr, there is a mechanism to take into account these multiple sentences and obtain a better evaluation. The system does not include an internal way to do this, so the mean of the predicted scores are used for the ranking:

$$\mathrm{r}(P', U') \tag{4.4}$$

, where $P' = \frac{1}{M'} \sum_j^{M'} \gamma_{i,j,t} \forall i \in N', \forall t \in S$ and $U' = \frac{1}{M'} \sum_j^{M'} Ai, j, t | \forall i \in N', \forall t \in S$.

Table 4.4 contains the results for the SA dataset at a caption-level. Overall the later years are where the metric achieves the best scores. CIDEr, METEOR and BERTHA have similar correlations.

	VTT-16	VTT-17	VTT-18	VTT-19	VTT-20
BERTHA-SA	**8.1**	2.7	**9.1**	**7.5**	6.9
BERTHA-MA	5.0	2.8	3.3	7.2	3.2
BLEU-4	2.8	0.4	1.6	3.0	1.0
SentBLEU	3.6	2.2	4.3	5.3	2.7
CIDEr	1.7	**4.4**	5.9	3.5	10.7
Meteor	2.7	3.4	8.7	6.4	**11.5**
Rouge	0.2	1.7	7.6	4.6	9.3
SPICE	2.3	1.1	6.2	3.2	0.3

Table 4.4: Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment SA for TRECVid 2016 to 2020 participating systems; metric is trained on captions from all other (non-test) years

	VTT-16	VTT-17	VTT-18	VTT-19	VTT-20
BERTHA-SA	6.3	6.0	7.7	14.7	16.4
BERTHA-MA	4.2	6.6	5.8	**22.5**	**24.7**
BLEU-4	1.7	4.5	0.7	4.9	11.2
SentBLEU	2.4	3.5	1.0	-5.1	7.7
CIDEr	4.9	**8.3**	11.4	15.5	20.8
Meteor	1.7	**8.3**	**12.9**	22.2	23.5
Rouge	**6.4**	5.6	8.5	15.8	20.9
SPICE	1.3	0.1	1.2	1.8	0.1

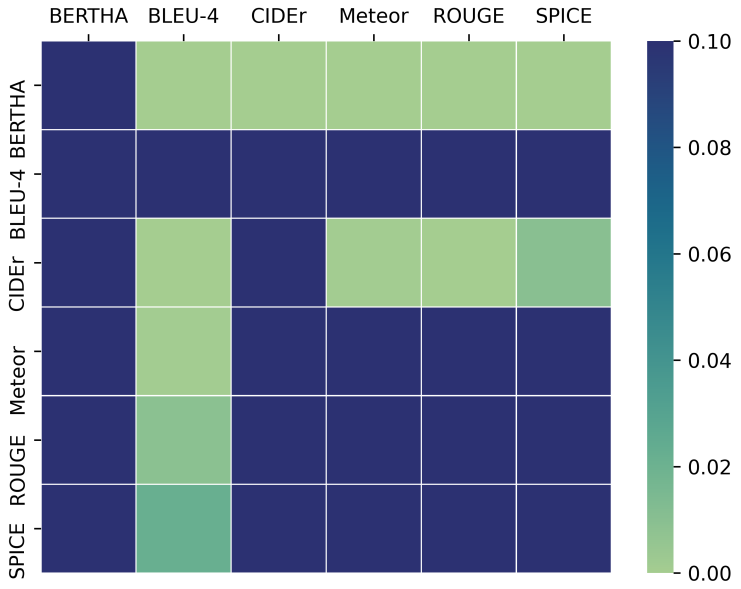
Table 4.5: Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment MA for TRECVid 2016 to 2020 participating systems; metric is trained on captions from all other (non-test) years

In the case of the MA dataset, we can see a similar behaviour as shown in table 4.5. BERTHA performs well and is the second or first metric with more similar performances to the human evaluation. Again the top performing metrics overall are BERTHA, CIDEr and METEOR without strong differences in the later years.

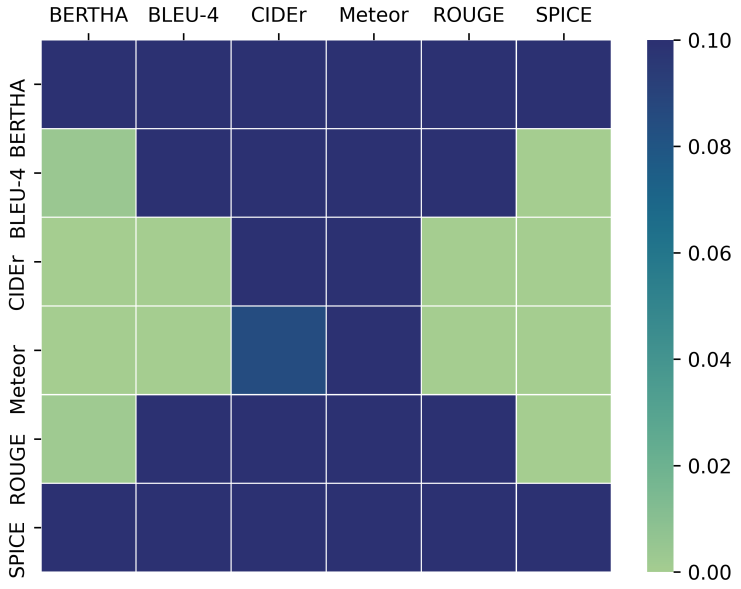
Similar to the system-level evaluation, the William significance test was performed to evaluate the correlations. Figures 4.5 show those results. BLEU-4 is the worst scoring metric in these scenarios. In MA, both BERTHA and SPICE do not achieve consistent results.

BERTHA outperforms the other metics in most of the test sets. In general terms, the MA dataset is the most complex dataset for BERTHA as it doesn't achieve good consistency, and the results vary depending on the year.

In general, we see a considerable drop in the correlation when comparing BERTHA



(a) William significance test of Video Captioning Caption-level Metric scores for the SA dataset



(b) William significance test of Video Captioning Caption-level Metric scores for the MA dataset

Figure 4.5: William significance test of Video Captioning Caption-level Metric scores with human assessment for TRECVid 2016 to 2020 participating systems. The p-value in a cell means the significance of the row vs the column in the absolute Pearson correlation.

trained in MA and tested in SA. On the contrary, the SA variant has slightly better results in the early years. However, in the later years of the task, this switches, and MA gets noticeable better results than other metrics. This means that the two datasets are not compatible, and the model can not be transferred between them in most scenarios. This is an important consideration as the same sentence can exist in both of them but have significantly different scores in each.

Using Multiple Human References

As mentioned before, in some years, the dataset has multiple references, up to five per video. This allows comparing each system generated caption with different human references. Notice that there is not a specific score per each human reference. It is shared between all the references to simplify the collection process. This is possible because the human references are close in terms of the human caption evaluation.

In the example in table, 4.6 the effect of using each of these references, named HUMAN-A, HUMAN-B, HUMAN-C, HUMAN-D and HUMAN-E, can be seen along with what happens when the five of them are combined. The same humans did not annotate all the years and this is why we focus on a single year for this experiment. As a single year does not have enough data, the training strategy is to perform 5-folds cross validation and show the mean. As BERTHA does not have any mechanism to use multiple references per video, the process to evaluate the five references per system caption is to compute the mean of the predicted scores before performing the correlation. For the other metrics, if they have any mechanism to take this into account, it is used; otherwise, the same approach is used as for BERTHA.

BERTHA achieves good performance even in the small dataset. As BERTHA is a learned metric it obtains better results than in the multiple years set up. This indicates that the metric is better suited when both the human evaluation and the system are more standardized. All the other metrics obtain worse numerical results than in the previous set up but they obtain similar ranking between themselves.

	HUMAN-A	HUMAN-B	HUMAN-C	HUMAN-D	HUMAN-E	5 Ref Captions
BERTHA-MA	**27.4**	**32.4**	**28.2**	**28.8**	**28.8**	**23.6**
BLEU-4	8.9	8.2	4.2	7.4	10.0	5.7
CIDEr	12.5	12.6	6.9	12.6	11.9	11.6
Meteor	12.3	14.5	9.5	13.6	10.7	11.7

Table 4.6: Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment MA for TRECVid 2020 participating systems train using a single HUMAN references

	VTT-16	VTT-17	VTT-18	VTT-19	VTT-20
BERTHA-MA	3.7	6.1	11.7	15.8	21.7
BLEU-4	0.2	4.2	4.7	0.2	0.0
CIDEr	5.9	8.1	12.6	17.0	22.2
Meteor	1.4	7.7	13.6	21.8	24.2
ROUGE	-8.8	3.6	4.8	11.6	16.6

Table 4.7: Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment MA for TRECVid 2016 to 2020 participating systems permuting the sentences evaluated

In terms of analysing the humans, HUMAN-C produces the worst outcomes for all the other metrics apart from BERTHA. This can explain some differences between each human annotator, which in the end means that BERTHA does not perform that well in the five-captions scenario. The main reason why the model can suffer in this set up is because it is trained in a dataset where different pairs of system generated captions and human references share the same caption and score, which can produce some level of confusion. All the other metrics get an average score in that case as they aren't retrained.

Words Shuffle in Evaluation

In table 4.7 we test what happens to the metrics when we train in a typical setting, but the sentences had their words randomly shuffled during evaluation.

As expected, the metric most affected is BLEU, which obtains a correlation near or equal to zero in some years. All the others, including BERTHA, are consistent with the previous experiments and only exhibit minor variations.

	VTT	MT	VTT+MT:VTT Test set	VTT+MT:MT Test set	VTT+MT:Mix Test set
BERTHA-MIX	18.6	38.0	17.3	38.0	15.8
BLEU-4	1.8	30.6	2.8	29.9	5.7
CIDEr	9.1	39.8	10.4	38.8	9.9
Meteor	9.2	46.9	9.8	45.4	11.5
ROUGE	5.8	41.3	5.5	41.8	7.7

Table 4.8: Pearson Correlation of Video Captioning and Machine Translation (MT) Caption-level Metric scores with human assessment for TRECVid SA 2016 to 2019 participating systems; VTT (TRECVid Video To Text) is only using the original data; MT is using only data from WMT [Ma et al., 2019]; VTT+MT:VTT Test set is train in all the data from both sets but test only in data coming from VTT; VTT+MT:MT Test set same train as the previous set up but test in only in MT data; VTT+MT:Mix Test set same training as the previous sets and test in a mixture of both data.

Including Machine Translation Data

In table 4.8 a study using external data not related to TRECVid is shown. In this case, data from Conferences on Machine Translation (WMT) [Ma et al., 2019] Metric task 2015 and 2016 is used, which is a task where an algorithm needs to learn to evaluate machine translation systems from human judgement. We include all the data translated to English, as both the references sentences and the system-generated ones are in English. In all the cases presented, the data is divided into 70% train, 15% validation and 15% test. The Machine Translation task is more accessible as it asks for a word by word translation, so both sentences, the reference and the system generated one, are very similar, so classical n-grams metrics perform well. This does not happen in a video captioning scenario and this explains the significant drop in the scores. BERTHA obtains the worst results when adding the machine translation data to the TRECVid data. This is noticeable in the mix set as it produces the worst results. The effect of adding TRECVid to the train and test only in MT data is noticeable.

This experiment demonstrates that the complexity of evaluating video captioning systems do not correspond well with a similar task like evaluating machine translation systems.

	No Human	Human train and test	Human only in train	Human only in test
BERTHA-MIX	-5.0	24.3	3.3	-1.4
BLEU-4	0.9	6.3	0.9	6.3
CIDEr	4.0	16.7	4.0	16.7
Meteor	5.4	18.6	5.4	18.6

Table 4.9: Pearson Correlation of Video Captioning Caption-level Metric scores with human assessment for TRECVid SA 2018 participating systems; showing four configurations; No human at all, human captions as a system in both train and test, only in test and finally only in train.

Including Human Captions as a System

One interesting aspect of this dataset is that there is a system that consists of only human captions. For the captions of these systems, there are also human judgements as in the other systems. In this case, the person in charge of evaluating these captions was not told that they came from another human.

Table 4.9 shows the results of a study on the effect of considering this human system in the different configurations. The study was performed on the data from TRECVid SA 2018 with a split of 70% train, 15% validation and 15% test. The most beneficial setup is to train and test with the human as a system. However, this gives an unfair advantage as it can learn what is considered a perfect caption. At the end of these experiments, it was decided only to use the no human configuration as it gives the best setup to compare each metric as it is the fairest in terms of actual data seen by the model.

4.4.7 Qualitative Analysis

Apart from the quantitative analysis presented above, a qualitative analysis is also interesting to understand how the model performs. To prepare this analysis, we use the concept of a word cloud to see which words are more relevant for each metric. The word cloud consists of the top 10 scoring pair of references-system generated captions. Each sentence is divided into words. Without performing any tokenisation, the words are used to compute the cloud. The cloud represents the most mentioned word as the most prominent word in terms of text size. The only words removed

from this plot are the common stop-words that do not add any information, words like “in”, “us”, or “a”. In figure 4.6 all the word clouds for the metrics are presented for the year TTV-20 of the MA dataset. It can be observed that metrics based on n-grams share similar behaviour and give importance to similar words.

Analysing the results, the first thing to notice is that all the metrics give some importance to words like “man”, “young”, or “white”. Those are the standard topics where a man or a young person is involved in some activity. Except for BERTHA, all the other automatic metrics give a considerable relevance to “dancing” and “shirt”, which is not relevant for the human assessment. BERTHA focus on “holding” and “black”. The other metrics fail to include “talking” as the main activity that humans consider necessary to differentiate good from bad captions.

This analysis helps to spot weaknesses in what is important for each metric as compared to the human assessment. In general, metrics that use only the intersection of words between the references and the system generated sentences perform poorly compared to those looking for tendencies or synonyms, which is to be expected as the task is to describe a topic and not a translation with exact words. BERTHA focuses on some different patterns than the other metrics and gets a high correlation. This implies that BERTHA is an excellent metric to use to complement the other ones.

4.4.8 BERTHA as a loss function

The final experiment with this new learned metric is to try and use it as a loss function. This function fullfill all the conditions to be use as a loss without needing to change anything.

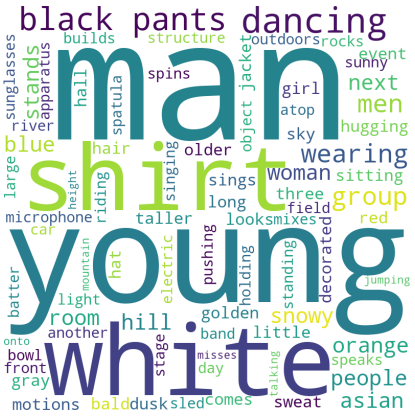
For this experiment we use the model propose in [Zheng et al., 2020b] as the full pipeline is available online [6] and it has the particularity that the loss function is a convention of the detection of the subject, verb and predicted together with a metric, like CIDEr, using reinforcement learning. Then BERTHA can be use as any

[6] https://github.com/SydCaption/SAAT

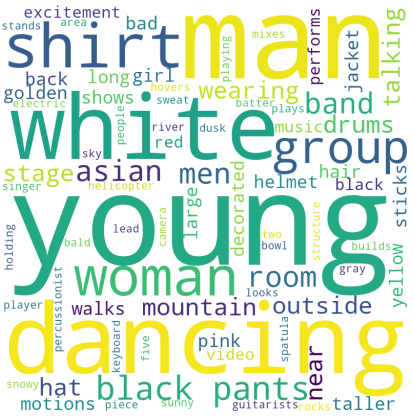
(a) Human assessment

(b) BERTHA

(c) Bleu-4



(d) CIDEr



(e) Meteor

(f) Rouge

(g) SPICE

Figure 4.6: Word cloud for TRECVid 2020 MA for the references level

	Initial loss value	Final loss value	BERTHA	CIDEr	BLEU-4	METEOR
BERTHA	9.3	9.2	0.4	0.0	0.0	0.1
CIDEr	8.2	0.8	0.5	47.1	38.9	28.2
Combination of BERTHA and CIDERr	5.4	4.5	0.5	31.8	25.3	22.4

Table 4.10: Comparison of training [Zheng et al., 2020a] using CIDEr vs BERTHA. The stats use to compare are: initial loss, final loss and final results in training.

or both part and use to train the same model.

We use only the data from TRECvid 2020 and the BERTHA-MA. To compare CIDEr use with the basic procedure of [Zheng et al., 2020b]. In the case of BERTHA it is used as the second part of the loss function without using reinforcement learning. All the other hyperparameters are set to the same values, the ones shared online. The results are included in Table 4.10. From the statistics of the loss, we can see that BERTHA is not able to guide the model in the initial steps so it can't progress. It is also possible that BERTHA doesn't produce a strong signal making it not suitable as a loss. There's a possibility that BERTHA lacks potency as a loss signal altogether. In our conclusive experiment, we attempted to merge the two metrics by averaging their losses after independently normalizing their values within the batch. Despite yielding promising outcomes, it's evident that CIDEr emerges as the dominant signal in this scenario.

4.5 Conclusions

In this Chapter, we propose a method to train a novel evaluation metric for video captioning. The technique learns to compare system-generated captions with human references from human judgements. We study the robustness in a real challenge scenario. The dataset used contains human assessment for system-generated captions. We detected some inconsistencies with the one used in the workshop, as only using a single annotator adds uncertainty. A more robust approach is using 15 annotators per caption to standardise the final score. BERTHA obtains a better correlation than the most commonly used metrics at the caption level. From qualitative analysis, BERTHA focuses on different topics so that it can be used to complement the

other metrics or as a standalone metric. BERTHA is purely a deep learning model, so it is easy to include in different architectures.

We prove that the learned metric can be trained in the TRECVid dataset to evaluate across the different years. The approach establishes a baseline which can be extended. These new waves of metrics have the advantage that they do not need a complex design to perform the evaluation and have a certain level of flexibility compared to metrics based on specific formulas. In general, the desired metric can be used across different datasets. However, human judgements could depend on the dataset; for instance, in a cooking video, a clear description of the process is required and typically not the kind of clothes the chef is wearing. On the contrary, in fashion news, we are more interested in the description of the dress than the actions performed. Traditional metrics rely on the wording used by the human references to evaluate these nuances; however, they don't explicitly account for them. In the cases of the learned metrics, this could be enforced in the training procedure without a significant change in the metric. One big problem with the traditional evaluation procedure is that they are somehow easy to fool. Learned metrics have the advantage in a black box scenario; they aren't as easy to cheat, providing a more robust evaluation.

The most prominent criticism of this method is the need for the training process, which requires a certain amount of annotated data and computation. In the following chapter, we delve into how Contrastive Learning can enhance our results. Additionally, we investigate whether reducing the requirement for annotated data is feasible.

Chapter 5

Improving the model using Contrastive Learning

5.1 Introduction

In this chapter, we explore different techniques to improve the metric presented in the previous chapter. Compared to the text similarity metrics, BERTHA yielded promising results. However, the training methodology is simplistic and restricts the model's ability to learn.

One of the main observations in the previous chapter is that each year of TRECVid has a different distribution of samples in terms of quality. In the latest years, the overall fluency and descriptiveness of the systems are better than in earlier rounds. These give a variety of samples that can influence the model's performance depending on the focus. BERTHA can learn to evaluate incorrect captions but has more difficulty distinguishing between the correct ones. One way to solve this problem is to avoid overfitting and focus on learning to group the captions.

A methodology popular in the literature is contrastive learning. It is based on creating embedding spaces where similar samples are close and different ones are further apart where the concept of similarity is defined as needed. It can be a classical definition pertaining to samples coming from the same class or from

similar inputs. In our case, however, we define similarity in a different way; we want to measure how similar the system output is to our set references in a way that imitates human assessment.

In this chapter, we investigate this and show that whilst improvements are possible by adopting this approach, they are marginal in nature as the model tends to overfit the training set as a result of a lack of data. We show that augmentation techniques provide some gain but do not achieve the same level of performance as when using supervised learning. Contrastive learning is promising but very dependent on the number of samples in its base formulation, and it aims to learn without labels. This means that the techniques need a more extensive set of samples from our population.

This chapter starts with a short introduction to contrastive learning and a presentation of similar works, Section 5.2. Following this, the proposed method is described in Section 5.3. The results are presented and discussed in the next Section 5.4. The chapter concludes, Section 5.5, with a summary and some thoughts on how this approach could be improved.

5.2 Related work

Contrastive learning is a training process which aims to bring together similar samples within the feature space and push away different ones. The most common approach is to use an augmented version of the sample as a positive instance and use other examples in the dataset as negatives.

The basis of this technique is contrastive loss which can be defined in multiple ways. The basic definition is the following [Chopra et al., 2005]:

$$\mathcal{L}(x_i, x_j, \theta) = \begin{cases} \|f_\theta(x_i) - f_\theta(x_j)\|_2^2 & \text{if} y_i == y_j \\ max(0, \|f_\theta(x_i) - f_\theta(x_j)\|_2^2) & \text{if} y_i != y_j \end{cases} \tag{5.1}$$

where x refers to the input vector, y are the labels and $f_\Theta(x)$ represents the model

being trained.

A more popular loss is the triplet loss [Schroff et al., 2015]. This loss is based on using a positive sample x^+ and a negative sample x^- to learn the distances. This method strongly relies on appropriate negative samples being available as it only sees one each time.

$$\mathcal{L}(x, x^+, x^-) = \sum_x max(0, \|f_\theta(x_i) - f_\theta(x_i^+)\|_2^2 - \|f_\theta(x_i) - f_\theta(x_i^-)\|_2^2) \tag{5.2}$$

To address the problem of selecting relevant negative samples, the triplet loss can be extended to use multiple negative n [Sohn, 2016]:

$$\mathcal{L}(x, x^+, (x^-)^n) = log(1 + \sum_z^n exp(f_\theta(x)^T f(x_z^-) - f_\theta(x)^T f(x_z^+))) \tag{5.3}$$

One type of loss that has gained significant traction in the field is InfoNCE which is used in CLIP [Radford et al., 2021b]. Its basis is NCE (Noise Contrastive Estimation) [Gutmann and Hyvärinen, 2010], a method for estimating parameters. The technique uses logistic regression to split a noise distribution from the data distribution. The original compares only one positive sample, x, and one noisy instance, x', but it can be easily extended to multiple noisy examples. InfoNCE builds on top of this and aims to find the positive sample inside a set of noisy samples. If we consider p_θ the probability of the sample being a positive sample and g the probability of the sample being noisy, NCE is formulated as:

$$l_\theta(u) = logp_\theta(u) - logg(u) \tag{5.4}$$

$$\mathcal{L}_{NCE}(x) = -\frac{1}{n}\sum^n[log\theta(l_\theta(x_i)) + log(1 - \theta(l_\theta(x_i'))] \tag{5.5}$$

$$\tag{5.6}$$

This formula can then be used to define InfoNCE from a context vector c and

$f(x, c)$ as an estimation of the density ratio:

$$f(x, c) \propto \frac{p(x\|c)}{x} \tag{5.7}$$

$$\mathcal{L}(x_i, c) = -\mathbb{E}[log\frac{f(x_i, c)}{\sum_j f(x'_j, c)}] \tag{5.8}$$

This loss is actually more relevant in the context of the work in the next chapter, where it is used together with CLIP.

As mentioned, sample selection is very important when using contrastive loss. Multiple strategies can be explored to select samples. All the following architectures follow a similar format: they use "two" encoders, one to encode the query and another to codify the key or keys. Query refers to the value that we want to store and the key is the value used to search for that query in the "dictionary". Then both vectors are used to compute a similarity metric, usually cosine similarity, to observe the distances between the samples.

The first architecture to explore is the end-to-end approach [Jiang et al., 2020]. In this case, both encoders are trainable, and the strategies are to train them at different times to ensure diversity in the representations. The encoder for queries only sees the original samples, while the encoder for keys sees both augmented versions of the positive examples and negative ones. The most significant disadvantage of this method is that the batch size determines the number of negative samples and large batches suffer from optimisation problems [Goyal et al., 2017].

The so-called Memory Bank [Misra and Maaten, 2020] is a module that can be added to the architecture to avoid this problem. The Memory Bank stores the feature representations of the previous sample seen in the prior epoch. This representation is an exponential moving average of the features. The disadvantage of this method is that the encoded vector needs to be updated every few epochs, which implies an overhead in the computation.

An improvement to this idea is presented in [Chen et al., 2020a] where the authors

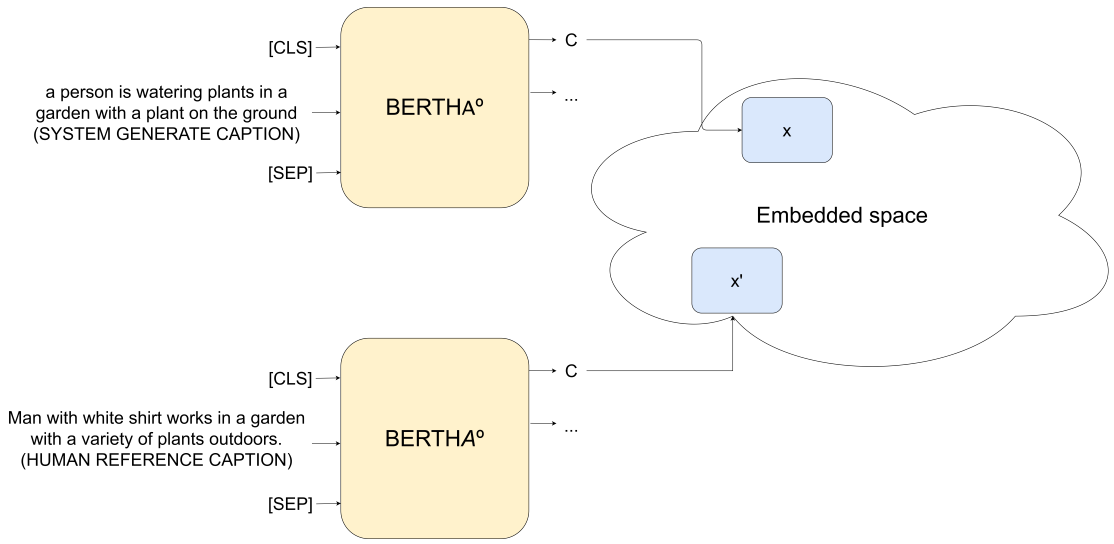


Figure 5.1: Representation of the evaluation of a system-generated caption against a human reference. The same model, BERTHA, is used to predict both embed vectors, x and x'. The distances, e.g. cosine distance between the two vectors, determine how similar are the two captions. A closer caption means that is more similar than a further one.

replace the Memory Bank with a queuing system called a Momentum Encoder for Contrastive Learning or MOCO. The basic idea of MOCO is to cycle the previous batches as negative samples and, in that way, limit the memory use of the model. Using this method does not allow for updating the encoder for keys. The solution they apply is to use a momentum update from the query encoder. This is our main methodology to apply contrastive learning to our model. More details of the specific process are discussed in the next section.

5.3 Methodology

In this work, we explore contrastive learning to improve the evaluation process presented in the previous chapter, BERTHA. We use the same base model BERT (Bidirectional Encoder Representations from Transformers) [Devlin et al., 2018]. The basic architecture is the same as BERTHA 4, the main difference is that the output layer is increased from one output value to a vector, of dimension 256. To use well contrastive learning we need to define a high-dimensional space where the two vectors can be compared in terms of distances.

The idea behind this Chapter is to train a model which predicts a vector in an embedded space. This space should have the property that closer captions are more similar and distant ones are more different from each other. Figure 5.1 illustrates this idea with an example.

As this is a more complex problem and training is more costly, we decided to limit the dataset. For our basic setup, we use the Multiple Annotators TRECVid dataset. However, to make more comparisons, we also include MSCOCO (Microsoft Common Objects in Context) [Vinyals et al., 2016b].MSCOCO is an image captioning dataset for which both the human evaluations and the 2015 systems outputs are released. MSCOCO evaluation is based on 5 diferent scores:

- M1: Percentage of captions that are evaluated as better or equal to human captions.
- M2: Percentage of captions that pass the Turing Test.
- M3: Average correctness of the captions on a scale of 1–5 (incorrect - correct).
- M4: Average detail of the captions from 1–5 (lacking details - very detailed).
- M5: Percentage of captions that are similar to human description.

These scores are given only at the system level, so the comparison is more limited than in the dataset presented in the previous chapter 4.

We explore the inclusion of the MOCO module described previously. We use the improvement presented in MOCOv2. MOCO uses a queuing system which collects samples from the previous batch. However, it can be a different size than the batch. To simplify the experiments it is commonly desired to use a multiple of the batch size to both improve accessibility and computational speed. In our cases, we use 2 $\times$ the size of the batch in most of the configurations. To ensure that positive samples are not included in the memory, all the captions produced for the same video as the caption to be evaluated are removed from the computer of the loss.

Regarding replacement, we always follow the rule of first in, first out. As previously, updating these representations is difficult using only backpropagation. The original authors explain that simply copying the query encoder gives poor results as the representation needs to be more consistent. The proposed solution is to use a momentum update, which consists in updating the new key encoder with a combination of the previous one and the query encoder by a factor of m. The momentum coefficient, m, can explicitly define the similarity between the two networks . The improvement in MOCOv2 is using a longer training epoch, extra augmentation, and cosine learning rate. We adopt all of these aspects; however, in the case of data augmentation, we follow EDA [Wei and Zou, 2019] as MOCO uses an image as input, but we aim to evaluate text.

5.4 Results

In the first experiment, we look to finetune the base model using Contrastive Learning and MOCO in their basic configuration. Only human reference captions are used to train the model in these initial setups. The idea behind it is that all the human references for a single video should be as close as possible. Then when we need to compare against the system-generated sentences we could measure how close they are to the "human area" in the embedded spaces. As in the dataset, there are five human sentences per video. We train leaving one human sentence out as an anchor and using the other as positive examples. The training procedure overfits the training data as there are few examples in the entire dataset, and the samples are very diverse. The system-generated caption is also added and used as an anchor to boost the number of instances. To further generate more data, eda [Wei and Zou, 2019] is applied to all the anchor sentences. EDA apply the following transformations:

- Synonym Replacement: replace a word with one of its synonyms.
- Random Insertion: insert a random synonym of a random word in the original phrases in a random position in the sentence.

- Random Swap: swap the position of two random words in the sentence.
- Random Deletion: delete a word in a sentence.

All these transformations could produce sentences that are wrong or too far from the original one. But small changes even if the sentence is incorrect will help to build the desired spaces. In our experiments, we limit to only generating 10 new sentences for each original phrase.

In the experiment, we compare various scenarios. The first one uses only the augmented references. Although we have multiplied by 10 the number of data is still overfit early on. The next test is to add the system-generated captions, which we call hypothesis, we don't know if they are good or not but we know that they are sentences generated for the specific video. We expect a minimum of information which could contribute to distinguishing sentences built from one video from the ones generated from another video.

In figure 5.2, we can observe that the simple scenario of only using the references overfits early in training. In the data augmentation case with the system-generated ones also included, it also overfits, but by resetting the learning rate at around the 500^{th} epoch, the model diversifies the knowledge and starts to learn again.

For results, we compare against METEOR [Banerjee and Lavie, 2005] and CIDEr [Vedantam et al., 2015]. To evaluate the Correlation between the metrics and the human judgements, we use the Pearson Correlation. In this experiment, the model is trained on the data of 2016, 2017, 2018 and 2019 and tested in 2020 with results presented in 5.1 which shows an improvement when using more data even if we aren't sure about the quality of it.

When testing in MSCOCO, we can analyse in more detail how data augmentation affects the results. Table 5.2 shows the results for training with data of TRECVid 2016, 2017, 2018 and 2019 and tested in MSCOCO. Interestingly, only the augmentation yields a significant improvement in the scores describing the quality of the captions in terms of similarity to human captions (M1, M2 and M5). This shows that the data for TRECVid is very focused on evaluating the general quality

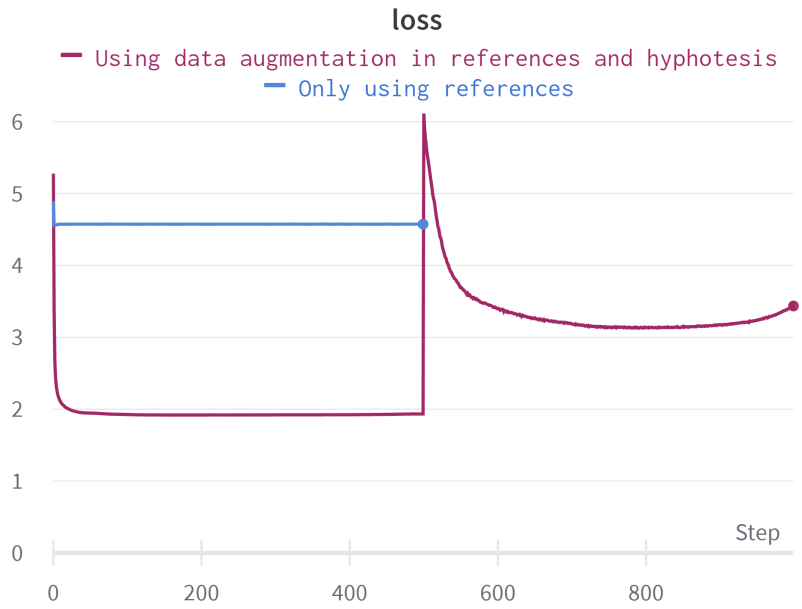


Figure 5.2: Example of loss in the training cycle when only using the reference captions vs using data augmentation in the system and human captions.

Table 5.1: TRECVid baseline and data augmentation. The metric is Pearson Correlation with human annotations at the caption level. CL refers to basic Contrastive Learning, aug is data augmentation, and ref-aug is referencing only data augmentation

	Pearson's
CIDEr	0.197
METEOR	0.234
Baseline CL	0.027
CL+ref-aug	0.080
CL+aug	0.111

Table 5.2: MSCOCO baseline and data augmentation. Metric is Pearson Correlation with human annotations at the system level. The definitions of Mx can be found in Section 5.2.

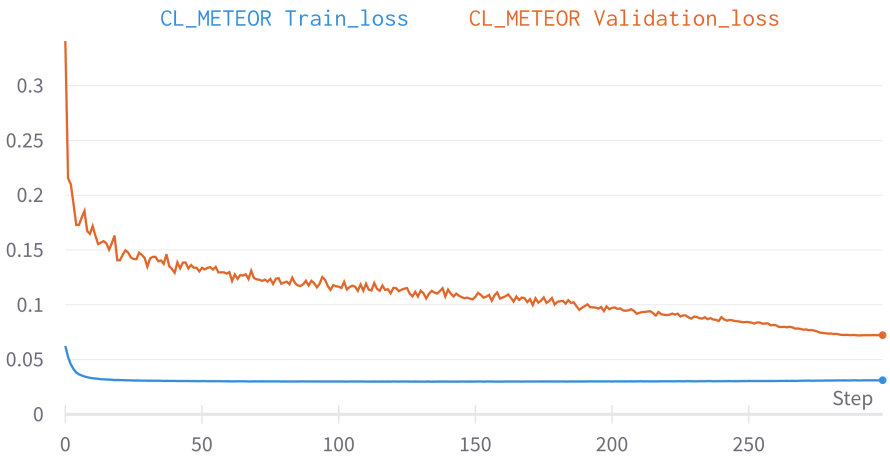
	M1	M2	M3	M4	M5
CIDEr	-0.447	-0.489	-0.380	-0.265	-0.497
METEOR	-0.365	-0.414	-0.408	-0.123	-0.470
Baseline CL	0.016	-0.008	-0.224	0.171	-0.057
CL+aug	-0.078	-0.103	-0.259	0.104	-0.172

of the caption and not other attributes of the captions.

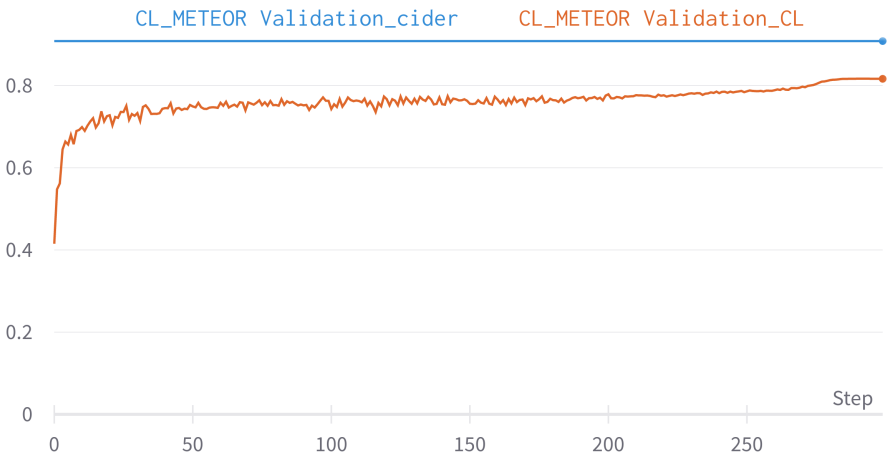
Not using labels has proven not to achieve an improvement compared to BERTHA. In the next set of experiments, we guide the model to match the scores given by METEOR. To do so, we enforce the constraint that the similarity between the positive sample and the anchor caption should result from computing METEOR between them. We continue using 2016, 2017, 2018 and 2019 and tested in 2020. From all the data used to train the model a small set of the 20%, without taking into account the year, is used for validation. The graphics in 5.3 show that the model can match those scores in training. Validation is 20% of the entire training set. In this split, the experiment also obtains a promising result. Although we achieved a 23.87% improvement in the test (TRECVid 2020), compared to previous experiments, the final result has a Pearson's correlation of only 0.1375 which is lower than all the other metrics.

5.5 Conclusions

In this chapter, we study the use of Contrastive Learning to improve the training procedure of models like BERTHA. This technique has proven to work in many different tasks [Hessel et al., 2021]. Although the results show no improvement, they demonstrate that this technique can be applied in the right circumstances. Our experiments lead us to conclude that more data (see table 5.1), and more diversity (see table 5.2), could produce the required improvement. This dependency can also be observed in the other task [Radford et al., 2021a]; CLIP is trained on a dataset of



(a) Loss in training and validation.



(b) Correlation between the human assessment and the predicted score and CIDEr

Figure 5.3: Example of loss and prediction in the training cycle when using METEOR as a guide for Contrastive Learning.

400 million training examples. Even with data augmentation techniques, we couldn't achieve these levels of data and keep the same level of diversity. There is also the problem of computational resources to train on those samples. Our approach is only text based, but CLIP has proven that image and text data can be combined in the same spaces. In the next chapter, we explore using the potential of Contrastive Learning to integrate visual and textual information in order to improve the video caption evaluation.

Chapter 6

Evaluation of Video Captions Automatically Generated Using Vision and Language Models

6.1 Introduction

Vision and language models have become extremely popular recently. One reason for this is that they are easily transferred to other tasks. In particular, they have been demonstrated to work well in the evaluation of automatic image captioning [Hessel et al., 2021]. This has made it possible to evaluate systems without the need for references or additional information apart from the image and the caption. However, to date, these models do not provide a straightforward way of evaluating video captions.

Although CLIP has already been used as a loss metric [Shen et al., 2021] there is another benefit to using it as the evaluation metric. The loss is only trying to evaluate how close is your output to your reference sentence or image, so this function needs to be able to point the model in the right direction while training. In the metric situation, the objective is slightly different as you want to measure how well your model is performing in general, and then use it to do another task like

model selection. So the metric needs to be able to evaluate the model and compare it against others after training. A good loss is not always a good metric and the other way around, if the function can be applied as the two.

In this Chapter, we propose using these models for video captioning evaluation. As previously discussed, video captioning is a similar task to image captioning evaluation; however, more factors need to be considered as the input contains more information than a single image. A video can be understood as a collection of individual images, the frames, using a single image ends with the problem of selecting what frame is the best one to represent the video. The most straightforward way is to consider the middle frame or a random frame as the representative for short videos. However, we don't know about the content of that frame; is it similar to the other frames? the image is focused? To evaluate the video is better to consider all the frames and see how to select the ones that are going to be included in the evaluation. The idea of this work is to study which strategies are the best in order to select a final image embedding to be compared using a vision and language model. We test two vision and language models against image and video captioning evaluation datasets to study their respective performances in a simple scenario of a single image and caption. The datasets used are Flickr [Hodosh et al., 2013] and Composite [Aditya et al., 2015] for image captioning and TRECVid 2016-2020 [Lebron et al., 2022] for video captioning.

The experiments reported here then explore different strategies to use video frames, from using a single frame as input to computing the mean of the embedded frames to a more complex experiment where the frames are clustered. Experiments demonstrate that using clustering methods to select a few frames to compute the final score gives an excellent correlation with human judgement. The finetuning of these models is also studied to further adapt them to the task. We obtain better results than the other metrics in various setups with the best results achieved using clustering and model retraining. Finally, the outliers and the human bias in the annotations could affect the quality of the evaluation, in particular when training

a new model. The final set of experiments attempts to filter human judgements to find the annotations that do not follow the same pattern as the others. We discover that removing only a small set of captions improves all the metrics tested.

This chapter is structured as follows: a review of progress in the field of learned metrics is presented in Section 6.2, this review is centred around the latest trend of using vision and language models in similar tasks compared to the classically used n-grams and their variations. Section 6.3 describes the methodologies to integrate the vision and language models in the evaluation pipeline for video captioning and how to use them with the frame information. Next, Section 6.4, shows and discusses the results in the video captioning dataset described in 4 and some image caption datasets. The chapter finishes with some conclusions in Section 6.5.

6.2 Related work

As discussed previously, traditionally video captioning and image captioning are evaluated by comparing the captions against references written by humans. The use of these reference sentences makes it necessary to have human annotations for all the captions in order to obtain the final evaluation. Furthermore, all traditionally used approaches fail to account for the visual information in the image.

The next evolution are learned metrics where a model is trained to evaluate the caption. For example, BERTHA, Chapter 4, use a language model trained to evaluate video captioning using only text data. These initial methods are limited by the lack of visual data in the evaluation process. An improvement to this idea is to include this information in various ways. For instance, TIGEr [Jiang et al., 2019] and ViLBERT[Lu et al., 2019] use pretrained models to evaluate an image caption using both textual and visual information in two different pipelines, which are fused later on. LEIC [Cui et al., 2018] uses a CNN and an LSTM to extract the visual and textual information and combines them to detect if the caption is written by a human, removing the need for human references. All of these ideas provided some advantages over traditional metrics, such as not requiring references and utilising

visual data. However, they tend to be challenging to train, and the correlation with human judgements is not significantly better than the previously mentioned metrics.

Recent developments have explored combining images and text in a shared space in more depth. These models are trained in very large image and caption pairs datasets using contrastive learning to find image embeddings directly from the associated language. The objective of these models, like CLIP [Hessel et al., 2021] and CLOOB [Fürst et al., 2021], is commonly focused on the task of zero-shot learning. This goal makes it easy to transfer the model without retraining. They have been used in other tasks like image search, OCR or action recognition with state of the art results.

Taking this idea, CLIPScore [Hessel et al., 2021] proposes various setups to evaluate image captioning using CLIP [Radford et al., 2021a], a well-known vision and language model. They propose using similarity between embeddings of the caption and the image to rank the scores. They obtain state of the art results in multiple images captioning datasets. Similarly, UNITER [Chen et al., 2020b] also proposes a vision and language model pretrained on pairs of captions and images, obtaining similar results to CLIPScore. These approaches prove to better correlate with human judgement than previous metrics and have the advantage that they do not need human references. Both of them perform well in image captioning evaluation; however, they are limited to only using a single image.

6.3 Proposed Method

6.3.1 Vision and Language Models

The two cross-modal vision and language embedding models we use are CLIP [Radford et al., 2021a] and CLOOB [Fürst et al., 2021]. They follow a similar structure: both of them have an image encoder and text encoder that is trained in huge datasets of image-caption pairs extracted from the web. Both use a Transformer as a text encoder, the difference being that CLIP uses a Vision Transformer as the image

encoder, while CLOOB is built based on a ResNet image encoder.

CLIP is trained to maximise the cosine similarity of the two vectors while minimising the similarity of the mismatched pairs using InfoNCE [Radford et al., 2021a, Oord et al., 2018]. CLOOB instead uses InfoLOOB [Fürst et al., 2021, Cheng et al., 2020], which aims to remove the "explaining away" problem of InfoNCE, which is the problem of focusing on a small set of features and discarding all the other relevant information. They also add Modern Hopfield Networks to reduce high variance [Fürst et al., 2021].

CLIP is trained on a dataset of 400 pairs coming from other image captioning datasets. CLOOB uses Conceptual Captions, which is a dataset consisting of 2.9M images-caption pairs.

6.3.2 Evaluation Methodology with a Single Image

Following the same methodology as [Hessel et al., 2021] we use the image and text encoder of the vision and language model and compute the cosine similarity between the two vectors.

$$s(c,t) = \cos(\text{VLM}_{image}(c), \text{VLM}_{text}(t)), \tag{6.1}$$

where c is the image, t is the textual caption and VLM is the vision and language model divided in the image and text encoder, and

$$\cos(u,v) = \frac{u^T v}{\|u\|\|v\|}$$

is the cosine similarity between embeddings u and v. In the most simple scenario, the setup is the same for image captioning and video captioning. In the case of video, only the middle frame is used. An additional advantage of this metric compared to traditional metrics is that it does not require human references to evaluate the caption.

This configuration can be extended to use multiple references. In this case, the caption is compared to the references, unlike in [Hessel et al., 2021] where the authors

compare the references to the image. Then the maximum similarity is added to the previous similarity score between the image and the caption:

$$\begin{aligned} s(c,t,R) = &\cos(\mathrm{VLM}_{image}(c), \mathrm{VLM}_{text}(t)) \\ &+ \max_{r \in R} \left\{ \cos(\mathrm{VLM}_{text}(t), \mathrm{VLM}_{text}(r)) \right\}, \end{aligned} \tag{6.2}$$

where R is the set of references.

6.3.3 Evaluation Methodology for a Video

In the case of a video, the visual information comes from multiple frames; however, none of the vision and language models have a mechanism to include all the frames simultaneously. One approach is to compute the similarity of the caption for all the frames and take the maximum similarity. The same idea can be applied with the mean and minimum similarity:

$$s(F,t) = \mathrm{op}_{f \in F} \left\{ \cos(\mathrm{VLM}_{image}(f), \mathrm{VLM}_{text}(t)) \right\}, \tag{6.3}$$

where $\mathrm{op} \in \{\min, \max, \mathrm{mean}\}$ and F is the set of frames. Videos are usually composed of a few shots, so it is unnecessary to compute the similarity between all the frames in the video. Some of these frames may be noise or a blank frame between shots, producing undesirable results in terms of the similarity with the caption. Moreover, some shots contain the majority of frames, which can affect statistics like the mean. To reduce the effect of these issues, we study clustering and classification techniques to detect similar frames. The first method we test is to find the k-NN [Fix and Hodges, 1989] frames to the caption and compute the similarity with them. In this method, we find the captions that are close to the sentence and use only those to compute the similarity. Using multiple frames mitigates the effect of possible noisy frames in the evaluation. It also only takes frames that are closely described by the caption and not any other random action that occurs in the video. The second method we test is based on clustering where we use k-means [Lloyd,

1982] to identify the different shots in the video and obtain a small set of frames that represent the main parts of the video. In these cases the caption is not involved in the selection of frames, the frames used are the most similar from the different subsets of frames, what is expected to be a small scene happening in the video. In both methods, the selection process of k is by trying multiple values in the datasets of the other years of TRECVid-VTT from 2 to 20 and selecting the one that gave the best performances.

6.3.4 Training Models

One reasonable step to continue with this line of work is to explore if the pre-trained model can be fine-tuned to the specific video captioning evaluation task. For this, we use the same loss functions as in each original paper. We keep most of the hyperparameters but change the learning rate and the weight decay to a lower one to speed up the learning phases of this particular task for the already-trained model.

For the data, we took all the TRECVid-VTT dataset 2016 to 2020 [Lebron et al., 2022] and reserved 2020 for testing, using the rest of the years as train/validation. During training, we found that using the predicted captions did not improve the final metric, which was discarded early, and only the references were used. In terms of the size of the dataset, only using four years of data for training does not provide enough pairs to train the model. To increase the amount of data, we augment the text. However, data augmentation is not used at the image level, all the video frames are used as separate image-caption pairs keeping identical scores and references. The primary data augmentation techniques used for text are substitutions, remove, insertion, swap and synonym replacement. The data augmentation strategy produces a low correlation in validation as there are plenty of pairs with close similarities and equal human scores.

6.3.5 Filtering Poor Human Judgements

In a dataset, there is always the possibility that one annotation does not follow the same criteria as another, even if it has come from a good annotator. Human error can also result in mistakes being included in the dataset. We design a simple way of filtering some of the poor human judgements.

Two commonly used correlation measures, Pearson and Spearman, are based on the assumptions of monotonicity in the scoring system, either in the actual score or in the ranks. Considering these premises, it is straightforward to find the line that fits the sub-data being analyzed. Considering a good metric, the expectation is that there is a relation between the human scores and the metric scores, and the mistakes in the ordering are the results of small changes in the scores. However, in that line, it can identify points that differ greatly from the premise of only small changes. Those outliers should come from the human scoring as the metrics are relatively more stable in how they give the scores for these specific datasets. Then there are various techniques to find outliers to this line. We use Cook's distance [Cook, 1977] as the measure to identify outliers. Cook's distance measures the influence of a data point on a fitted regression. A good rule of thumb[Cook, 1977] is to select four times the mean of all the distances as the distance at which a sample is classified as an outlier. We use this assumption to discard poor human judgements considering them as outliers.

6.4 Experimental Results

6.4.1 Image Captioning Evaluation

Our initial experiment is to test both models on image captioning datasets and in a video captioning one, but whereby only the middle video frame is used. Table 6.1 presents the results together with some of the most popular metrics.

The datasets used for image captioning are Flickr 8k [Hodosh et al., 2013] consisting of 17k human evaluations of video and captions pairs with scores between 1

	Composite τ_c	Flickr8k τ_c	VTT'19 r	VTT'20 r
BLEU-4*	30.6	30.8	4.9	11.2
ROUGE-L*	32.4	32.3	15.8	20.9
METEOR*	38.9	41.8	22.2	23.5
CIDEr*	37.7	43.9	15.5	20.8
SPICE*	40.3	44.9	-	-
TIGEr*	45.4	49.3	-	-
BERTHA*	-	-	22.5	24.7
CLIP	**54.7**	46.2	23.3	**37.2**
CLOOB	49.9	**56.1**	**30.1**	34.9

Table 6.1: Flickr8k and Composite Kendall's correlation with human assessment. TRECVid-VTT 2019 and 2020 Pearson's correlation with human assessment. Only using the middle frame to simulate an image captioning dataset. *results from [Hessel et al., 2021] for Flickr-8k and Composite and from [Lebron et al., 2022] for TRECVid-VTT 2019 and 2020.

to 4, and Composite [Aditya et al., 2015], which groups images from three different datasets: MSCOCO [Chen et al., 2015], Flickr8k [Hodosh et al., 2013], and Flickr30k [Plummer et al., 2015]. The final dataset contains approx. 4k image-caption pairs with four human evaluations per pair. Kendall's correlation is used in both datasets. For video captioning, the last two years of TRECVid-VTT are used. Each year consists of less than 1541 video-caption pairs from around ten systems. Each pair additionally has five references annotated by humans. The final human score is the standardisation of the scores coming from 15 different annotators. However, only a subset of the captions are annotated in this version. Only 1541 captions per year have human evaluations. Following on from the work reported in Chapter 4 we use Pearson's correlation to compare human judgement to the metrics scores. However, we switch to Kendall's correlation in further experiments as it is the measure most used in the literature. Table 6.1 presents the video captioning results.

Across all the datasets, both methods obtain state of the art results. Neither sufficiently outperform the other to declare it the best-performing one. Each one gets good scores in different sets. Comparing the other metrics, we can see that these methods obtain a better correlation than the classical metrics. Only TIGEr has comparable performance. We can conclude that either the visual information

	VTT'19 τ_c	VTT'20 τ_c
METEOR	16.1	23.5
CIDEr	13.9	20.8
CLIP_image	16.7	24.9
CLIP_similarity	12.4	24.6
CLIP_similarity(prenormalize)	16.2	26.9
CLIP_knn	17.0	**27.2**
CLIP_kmeans	16.7	27.1
CLOOB_image	19.6	22.4
CLOOB_similarity	20.4	22.9
CLOOB_similarity(prenormalize)	20.4	23.5
CLOOB_knn	20.5	22.8
CLOOB_kmeans	**20.6**	23.4

Table 6.2: TRECVid-VTT Kendall's τ correlation with human assessment using frame data. Images is with a single image; similarity is based on computing an average between the frames

is needed to achieve the final score or that we need to compare the caption to the references in a space where the sentences are related in a visual space. The embedding spaces are not built solely from a textual corpus but also a visual one, so sentences are closer if they appear in similar images instead of just in the context of a paragraph.

6.4.2 Video Captioning Evaluation

In this section, we discuss the application of this methodology in the video captioning scenario. The first thing to note is that none of these metrics has an obviously correct way to be applied in video evaluation. From analyzing individual videos, we observe modest differences in the similarity between the caption to other frames in the video. Table 6.2 compares the different ways to include this information. There are some explanations of why there is these differences in the correlations between VTT'19 and VTT'20. First, half of the videos in VTT'19 come from Twitter Vines while the rest come from V3C (Vimeo Creative Commons Collection) [Awad et al., 2020]; in the case of VTT'20 all of them come from V3C[Awad et al., 2021]. This second dataset has more depurated videos without unrelated segments inside the

	VTT'20 τ_c
Trained CLIP	27.6 (+1.4%)
Trained CLOOB	22.9 (-2.5%)

Table 6.3: TRECVid training using TRECVid-VTT 2016 to 2019 as train/validation and test in 2020.

short video used for the video-to-text task. The second is that the participating systems are different, these could explain some differences in the sentences. Also the number of points used to compute the correlations is relatively small. In general, we didn't identify major differences in the vocabulary or the videos from V3C. And similar differences can be observed in Chapter 4.

The differences between mean/min/max are not included because they obtain similar correlations, all of them are represented as the similarity result. The similarity and prenormalize experiments have the same configuration except that the feature vector is taken either from the L_2 normalization layer in the VLM or before this layer. A significant improvement can be observed from using the average prenormalized features, likely because the L_2-normalization changes the shape of the spaces modifying the distances between the image and caption.

From the clustering experiments, we can see that k-NN and k-means perform similarly in these short videos, and they both obtain some of the best results. This result means that removing noisy frames from the video can improve the evaluation methodology and that the whole video can be summarized in a few keyframes. k-NN also shows that focusing on evaluating a few frames and measuring how the similarity changes leads to a better evaluation. This implies that only using a few frames to perform the evaluation is necessary and that the difficulty is in finding these frames.

6.4.3 Training Results

One way to increase the performance in this task is to retrain the models with data from the task. In our setup, we use TRECVid-VTT 2016 to 2019 [Lebron

et al., 2022] as training and validation data and test in TRECVid-VTT 2020. The text data is augmented using substitutions, remove, insertion, swap, and synonym replacement adding 10 new pairs from the original ones. We only use augmentation on the text, not on the image and keep the same score as the original pair.

The model is trained between 100 to 1000 epochs, with Adam using a learning rate of 10^{-8} and a weight decay of 0.001. The model with the highest Kendall's correlation in validation is used to test. The experiments took from 1 to 3 days to run on an NVIDIA RTX 2080TI.

Table 6.3 shows the final results for the best CLIP and CLOOB models. We see an improvement in CLIP while CLOOB does not perform as well as the original. This may be because the loss function, InfoLOOB, is more complex and may need additional tricks and data to perform well.

6.4.4 Filtering Human Annotations

In Table 6.4, we observe the effect of removing outliers from the annotations. We analyse both the percentage of captions and the improvement across the metrics. Table 6.4 shows the results obtained removing outliers detected them using the actual model predictions and human judgements, Filter linear, or the ranks of both metrics, Filter ranks.

All the results are positive except for one case with CIDEr. Assuming that the scores should follow a linear pattern and removing the outliers based on this assumption, we obtain a performance increase between 10% to 20% when only removing 6% to 7% of the captions. Assuming that the score should be linearly correlated with annotator scores in rank, we obtain substantially better results, achieving an increase of up to 30% in the metrics whilst only removing 4% of the annotations. These experiments demonstrate that there are captions that penalise the correlations, meaning that these scores are not annotated following the same rules as the others and are outliers.

In Figure 6.1, we can see an example of a video and caption marked as an outlier

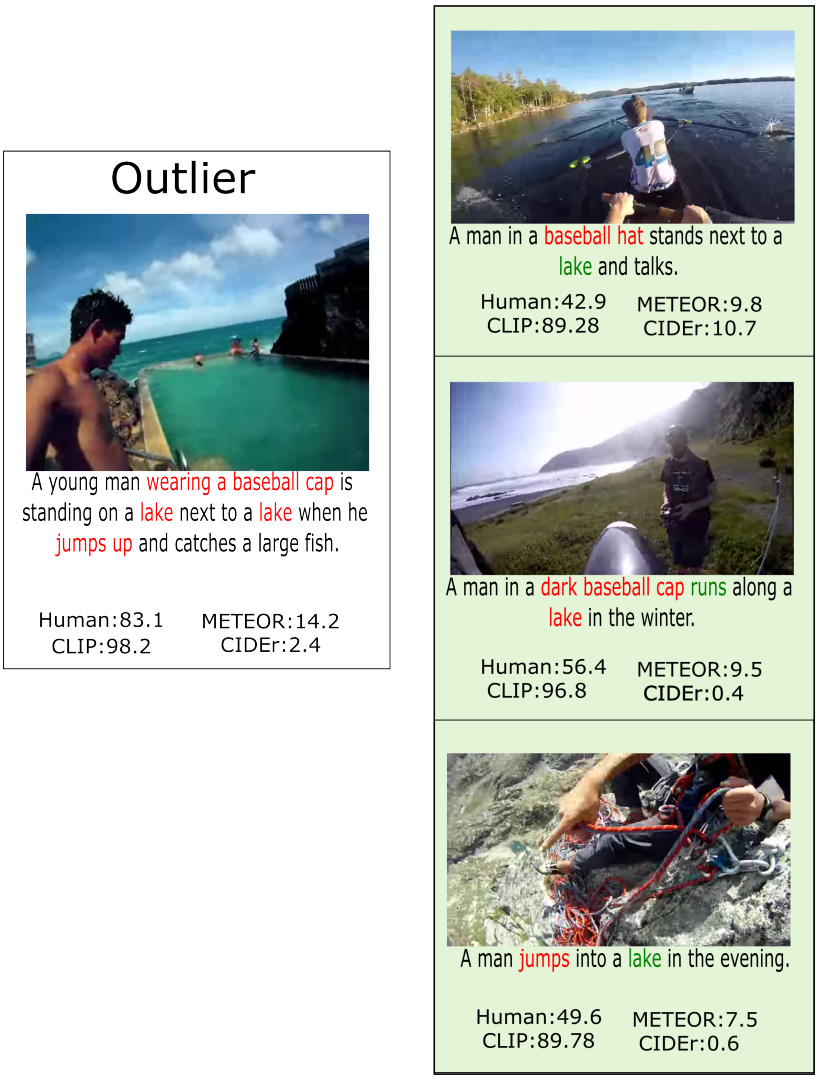


Figure 6.1: Example of a video caption removed by the filtering. The middle frame with the caption and its scores are on the left. There are three examples of similar video-caption pairs on the right that have different scores. The words we consider incorrect are represented in red and in green are the ones that the captions get correct, but the outlier does not.

with three similar videos which were not removed. We can see that the one removed contains many errors; however, all the metrics, including human judgement, give it a high score, while the other has a similar scene related to water and people but obtains worse scores. This relation to other captions could explain why it was filtered as an outlier.

It can also be observed that the best filtering comes from using the ranks, which

	VTT'20 τ_c	# captions remove
Filter linear CLIP	27.5 (+1.1%)	107 (6.9%)
Filter linear CLOOB	25.6 (+8.9)	93 (6%)
Filter linear CIDEr	20.7 (-0.4%)	86 (5.5%)
Filter linear METEOR	23.2 (+20.8)	106 (6.9%)
Filter rank CLIP	32.2 (+18.3%)	57 (3.6%)
Filter rank CLOOB	28.8 (+22.5%)	59 (3.8%)
Filter rank CIDEr	27.3 (+26.9%)	61 (3.9%)
Filter rank METEOR	25.3 (+31.7%)	56 (3.6%)

Table 6.4: TRECVid filtering captions based on linearity in the scores and on linearity in the ranks. Results use Kendall's correlation with human assessment and the percentage increase with respect to the baseline. Also shown are the number of image-caption pairs removed and the percentage that this represents of the total number of samples.

leads us to surmise that the evaluation score of the annotations follows a more complex pattern than a simple line.

6.5 Conclusion

In this chapter we propose a new way to evaluate video captions using vision and language models. These models have proven to work in very different scenarios as they provide a common embedding for two complex sources of data: images and text. One advantage these models offer is that they can accomplish the evaluation task without human annotations; however, the best performance is obtained using these references.

One problem with these models is that they do not include a way to add multiple video frames. To address this problem, we propose to use k-means or k-NN to find the best frames for the evaluation process. We retrain the model to better adapt it to the sentences used in video captioning. Our experiment shows that it is better to use the pre-normalised features of both encoders to compute the similarity. This reinforces the problem that the score obtained by calculating the cosine similarity is not as straightforward as the classical metrics. It depends on the dataset and the

models, and we can't assume a correlation between two separate data sets. Finally, we also noticed some outliers in the datasets. Filtering these improved all the metrics without removing too many pairs.

Exploring better ways to include video content could result in an improvement in the metric. Initial investigations could include detecting relevant frames and/or adding the motion information to evaluate the captioning containing some action verb. The methods proposed to detect outliers could also be improved by explicitly detecting human bias. However, to study the motivation of the annotators, a dataset is needed where the procedure is either more strict so that the specific aspects can be determined by the questions or by asking for some comments from the human annotators. Converting free text comments into valuable information sources is a complex problem. In the next Chapter, a new data collection process is introduced to first collect more fine-grained human judgements. Following this, instead of collecting comments, we ask annotators to edit the system-generated sentences, providing a more precise indication of where the model fails.

Chapter 7

Learning to Evaluate Video Captioning from Edits and Questions

7.1 Introduction

In the previous Chapter, we observed some inconsistencies in the dataset used. Other papers have found similar problems in other datasets [Howcroft et al., 2020]. A study of how well this performs in the image captioning domain is presented in [Hessel et al., 2021] along with some discussion on the utility of image captioning datasets for quality control. They point out that some of them, like MSCOCO [Vinyals et al., 2016a], a well-known dataset with human judgements for image captions, needs more data to be a good reference. The main problem is that filtering human annotations is a more challenging task than collecting them. Defining a good or bad sample is subjective and very prone to be influenced by the annotator's in-built biases. As such, in this chapter we present the design of a more complex data collection process divided into questions. The idea is to capture a set of aspects that help determine the quality of a caption. With this subdivision, a more fine-grain study can be performed.

Given the challenges with current approaches to video caption metric evaluation as discussed earlier in this thesis, in this Chapter, we present a new dataset of human judgments where we divide the general quality score used in the majority of datasets into five different concepts we consider relevant: quality/naturalness, how informative a caption is, the clarity of the caption, the usability and the presence of the main subject. We use the videos and systems' outputs of the TRECVid workshop [Awad et al., 2021] to collect human judgements. We also propose a new metric based on post-editing the system-generated captions. We compare this new metric to some of the most representative metrics used in video captioning evaluation. We show that it correlates well with human judgements and provides more information to find common mistakes in the systems.

The Chapter begins by discussing existing datasets for human judgments and methods for conducting quality studies. Following this exercise, Section 7.3 presents both the data collection process and the validation methodology used for our new dataset. Additionally, the new metrics for converting the post editing sentences into the score are presented. Section 7.4 then compares different popular metrics in this new dataset demonstrating its potential value to the video captioning research community. The Chapter finishes with some conclusions about the new dataset.

7.2 Related Work

7.2.1 Human Judgments Datasets

The most common way to evaluate metrics is to compare their correlation with human judgement. However, there are very few human judgement datasets and they are challenging to produce. Apart from resource issues, these datasets present additional challenges related to bias and other external factors that need to be taken into account, such as the detection of low-quality annotators or possible bias generated by the collection process itself. The validation process is also a difficult task, as it is necessary to filter bad-quality annotations without discarding many

of the good ones. This is especially difficult in borderline cases where annotations provide additional information compared to others, for instance, when an annotator wants to add a comment or explanation for the caption.

Some examples of human judgment datasets for image captioning are Flickr8k-Expert [Hodosh et al., 2013], Flickr8k-CF and Composite [Aditya et al., 2015], where the annotator was asked to grade the caption describing the image. CrowdFlower is another example where the question *is the caption describing the image?* is only binary but contains multiple judgements for each pair of images and video. A different example is MSCOCO [Vinyals et al., 2016a], where only system-level scores are provided and not for individual captions. However, they differ in the types of scores they provide. Five scores evaluate each system where the most used for image captioning systems evaluation is "M1: the percentage of captions that are evaluated as better or equal to the human caption" and "M2: the percentage of captions that are indistinguishable from human caption". A final type of dataset type is Pascal-50S [Xu et al., 2019], where the task is to compare pairs of sentences from humans or/and machines, covering multiple cases: human-human correct, human-human incorrect, human-machine and machine-machine.

An example of a dataset for video captioning evaluation is the TRECVid-VTT dataset, discussed in Chapter 4. Two setups were explored to conduct the assessment: using a single annotator per caption or a minimum of 15 annotators. Another dataset is [Shi et al., 2022b], where videos from VATEX were used to collect human judgements. In this dataset, only three human annotators were used. The main problem with this dataset is that only the general quality of the caption is studied with no feedback on why the sentences received a particular score.

7.2.2 Quality Evaluation and Criteria

Human assessment is a popular method to evaluate the performance of text generation. It can assess the sentences from either general or specific points of view. However, a problem arises when the evaluation process is divided into distinct as-

pects to be evaluated. The first significant difficulty is to define the concepts to be evaluated in a way that they are general, unique and non-overlapping [Al Sharou and Specia, 2022, Howcroft et al.,]. It is also challenging to design and implement a system to collect the required annotations quickly and precisely. The final step, validating the collection process's outcome, is also problematic. The most straightforward scenario is when data is collected from an expert, where it is only necessary to ensure there are no errors or mistakes. In this scenario, high-quality evaluations can be assumed. However, for speed and simplicity, many annotations are carried out using online services like Amazon Mechanical Turk, where people from different backgrounds produce annotations. In this case, the validation process needs to be robust as it needs to filter both bad-quality annotations and detect mistakes.

7.2.3 Post-editing evaluation

To analyse and score a system-generated sentence some works use a post-editing phase to correct major errors. Nowadays the most popular methods are based on using a large language model to rephrase the sentence and improve the description of some expected concepts [Li et al., 2024, Welleck et al., 2022]. This idea of looking at a better version of the sentence and measuring it post-editing distances has been already used in many text generation tasks like machine translation. This distance can be used to know how well the initial sentence is, for example, Levenshtein Distance [Levenshtein et al., 1966] or the posterior Word error rate (WER) measures how many character edits (insertions, deletions and substitutions) need to be done to convert a word to another. Another approach is to do something similar to what is previously mentioned in Chapter 6 to use the cosine distance in a common embedding space to measure the similarity between the two sentences. Using a human as a post-editor is the best way to obtain the best captions [Koponen, 2016], however, this is considered an impractical method to evaluate models because is time-consuming and difficult to define[Green et al., 2013]. These inspire us not to use this method to evaluate actual systems but to use them in the creation of a

dataset for metric evaluation which is more feasible as we only need to collect the data once.

7.3 Proposed Method

In this work, we focus our data collection on a subset of the concepts that define a good caption. We choose these concepts based on their relevance and how easy it is to describe them to an amateur data annotator.

We design our collection process based on the recommendations of [Howcroft et al.,] which provide some definitions for common concepts and give recommendations on how to design and report the human judgement collection process. In our approach, we decided on two methodologies to collect the data: through questions and through edits. In the collection process, we include five different questions which we consider to reflect the most relevant concepts required to evaluate a caption. Four concepts were selected from [Howcroft et al.,] and an additional question was to try to detect the complexity of identifying the main focus of the video. The concepts are quality, informativeness, clarity, usability and whether the actor/subject of the video is visible in the scene. The second method we apply is asking the human annotator to post-edit the caption to make it *correct* from their own point of view.

The first aspect we want to evaluate is its quality/naturalness. Defining the overall quality is a difficult task as many factors can influence this, such as grammar, clarity or accuracy. However, if we consider that a caption produced by a human is well-written and, in general, contains all the required concepts, we can then define that a good sentence is one that seems to be written by a human.

Whilst ideally, this concept should be sufficient as a measure of quality, we also have to consider annotator bias. A caption could describe the general scene, but it could omit a (varying) degree of detail or centre the description on irrelevant information. For this reason, we measure the informativeness and clarity of the caption. The aim is to capture the degree to which captions contain the minimum

information that any human would expect to be included. Clarity on the other hand represents how much information can be discarded and still describe the scene.

Captions can also contain incorrect information that needs to be removed or changed, and this is the focus of the usability question. A high usability score means the captions produced contained useful information, for instance, information making it possible to distinguish two videos from each other, as opposed to generic statements. This is difficult to measure using a number or a simple response to a question. Such errors in captions are better represented by correcting the caption as opposed to providing a score. These processes are defined in the following subsection.

The last question is independent of the other ratings. In this question, the annotator is asked if they can see the person/main subject of the video. The aim is not to provide a direct evaluation of the caption but instead to detect if the model can locate the topic or if the mistakes are due to it being challenging to see the video's primary focus. For instance, in an egocentric video, the metric could falsely detect that the most relevant action is performed by something in the scene instead of by the person recording the video.

7.3.1 POES: a New Metric by Post-editing of a Caption

The above questions provide an initial method of quantifying the quality of captions. However, exhaustively capturing all aspects that influence caption quality would require a lengthy list of questions that would be challenging to design and time-consuming to answer. We, therefore, additionally explore a new way to evaluate the caption by asking human annotators to correct any errors in it.

In this task, we ask the annotator to perform the minimum number of edits to the caption to make it correct. This can be directly related to the question about usability. A caption with many errors should also contain more modifications in response to this question. Of course, this process can also help to detect common mistakes quite apart from using it as a new metric.

In order to study this evaluation methodology and be able to use it together

with other metrics we need to represent these post-edits with a numerical score. We propose ways to convert them into quantitative measures: counting character changes and counting word changes.

As presented previously there are plenty of methods to use these edits as a metric or a distance. However, again this is a difficult task to evaluate because in a more extended analysis, we need to include things like the purpose of the sentences and if it is preserved, what kind of change has been done (lexical, grammar, etc.) or how much of the original sentence is preserved. The simpler way, to give a initial estimation, to measure this numerically is to compute the differences, in characters and words, between the original caption and the post-edited one. There can be multiple edits in this dataset, so we use the average score between all the modifications. The formula for the final score $POES^{word}$ is the following:

$$S^{word} = [a_{i,j,z}], i \in N, j \in M, z \in T \quad where \tag{7.1}$$

$$[a_{i,j,z}] = \begin{cases} 1 & \text{if } orgcap_i == fixcap_{j,z} \\ 0 & \text{if } orgcap_i != fixcap_{j,z} \end{cases} \tag{7.2}$$

$$normS_z^{word} = \frac{\sum \max_{i|j,z} S_{i,j,}^{word}}{N|M} \tag{7.3}$$

$$POES^{word} = \frac{\sum normS^{word}}{T}, \tag{7.4}$$

where N is the number of words in the original sentence $orgcap$, M is the number of words in the z edited captions $fixcap$ and T is the number of edited captions.

Using character differences, $POES^{char}$, is a more detailed analysis as it is less punitive on small changes like plurals or changes of tenses. The scores reflect the number of character changes between each word in the sentences. To normalise the scores we use the minimal length of each word. We also add a penaliser for a big change in the length of the words. The mathematical expression is as follows:

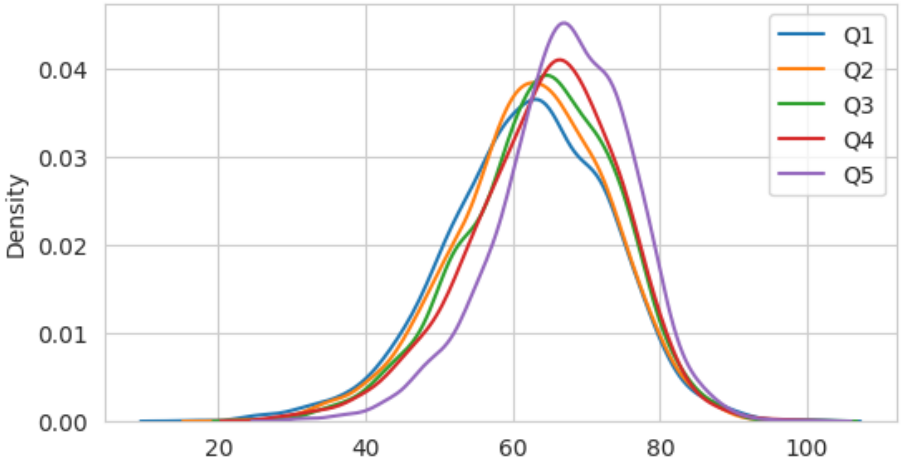


Figure 7.1: Density plot of TRECVid 2019 human evaluation ratings without quality control sentence

$$S^{char} = [a_{i,j,z}], i \in N, j \in M, z \in T \quad where \tag{7.5}$$

$$[a_{i,j,z}] = \text{norm}(L,K) + \frac{\sum_{i,j,z} b_{i,j,z,l,k}}{\min(L,K)} \tag{7.6}$$

and (7.7)

$$[b_{i,j,z,l,k}] = \begin{cases} 1 & \text{if } orgcap_{i,l} == fixcap_{j,z,k} \\ 0 & \text{if } orgcap_{i,l} != fixcap_{j,z,k} \end{cases} \tag{7.8}$$

with l being the size of $orgcap_i$ and k being the size of $fixcap_{j,z}$

$$normS_z^{char} = \frac{\sum \max_{i|j,z} S_{i,j,}^{char}}{N|M} \tag{7.9}$$

$$POES^{char} = \frac{\sum normS^{char}}{T}, \tag{7.10}$$

where L, K are the size of the word in the original sentences and in the post edited one respective way.

7.3.2 Collection Process

All the videos and captions used come from the TRECVid-VTT task from 2019 and 2020, presented in Section 4. We select a subset of videos and captions to work with by taking a random sample. In this subset, all the video captioning systems are represented by the same number of caption and video pairs and will therefore include a range of high-quality and low-quality system-generated captions.

To reduce bias in the evaluation, questions of interest are reformulated as *Likert* statements. The human annotator is then asked to specify the extent to which they agree with the statement on a continuous rating scale of 0 to 100. This allows the rating scale to remain constant for all criteria. The numerical range is selected as one that most people can relate to and give us enough detailed information. The final set of questions is formulated as follows:

Q1 This video description is written by a human.

Q2 This video description is detailed enough.

Q3 All the information in the video description is relevant.

Q4 All the information in the video description is correct.

Q5 The main person/actor/subject is the main focus of the video.

This simple way of presenting the evaluation criteria works well in our scenario, maximising the likelihood that the annotator clearly understands what is being evaluated. The continuous scale helps to standardise the final scores as it can be easily represented as a rating distribution.

As described before, we also include the task of editing the captions to make them correct. We divide the study into two levels. First, we ask if the sentence is valid, and second, if it needs to be fixed, the user is asked to make the minimum correction necessary so that corrections can be counted and used to represent the quality of captions. In this way, extensive editing indicates poor-quality captions while few edits indicate a good system.

Data collection was carried out using Amazon Mechanical Turk (MTurk). The MTurk service has been used in many annotation tasks because of its simplicity and capacity to quickly collect significant volumes of annotations. The process is divided into a subset of tasks where a random user, called a worker, will annotate a single subtask, known as a Human Intelligence Task (HIT). In our case, each HIT includes 40 videos and captions. This number is chosen empirically as the number of annotations from a single worker needed to validate the quality of their annotations. Figure 7.2 shows an example of the user interface for a single video. The process is as follows: the annotator first needs to look at the video and then decide if the caption is good enough. If not, a box will appear where they are asked to edit and correct the caption with as few edits as they see fit. Finally, they are asked to rate how much they agree with the questions (on a scale of 0 to 100). The reason to use the continuous numerical scale is to obtain a fine grain and give the freedom to the annotators to use whatever range they want. This process gives more details per user but adds the additional step of normalizing the final score of each user, which became more reliable if they evaluated many videos. In many other datasets, they use a 7-point Likert scale [Aditya et al., 2015] or a binary score[Xu et al., 2019]. In the literature of surveys, there isn't a clear best method, and each offers some advantages and disadvantages [DeCastellarnau, 2018].

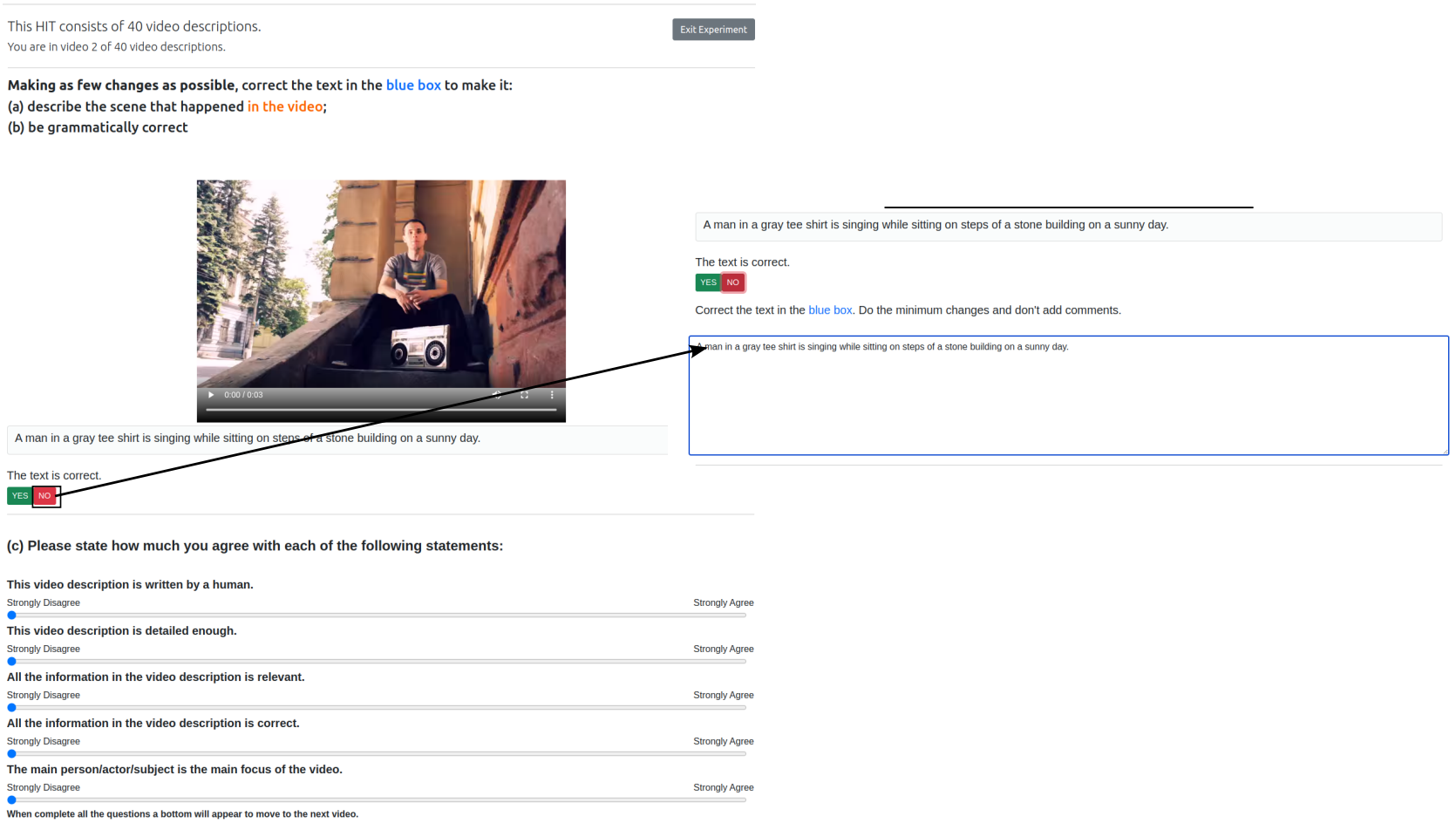


Figure 7.2: Example of the user interfaces used for collecting the annotations.

7.3.3 Validation

The main disadvantage of using Mechanical Turk is that it can include low-quality annotations, so a validation process needs to be applied. In our case, we design a process based on alerts where our system automatically detects suspected low-quality annotations, and an expert then decides whether or not to keep the HIT. The process additionally includes several sub-mechanisms aimed at identifying these noisy annotations.

The first step is to detect problems in answering the questions. To do so, an ANOVA test [St et al., 1989] is performed for the five questions expecting that, in general, the answer will have a similar distribution in each question. The second step is to count the number of edited sentences carried out by the user. As a rule of thumb, it can be expected that at least half of the sentences need at least a minor edit. The final step is to intentionally degrade the quality of the caption and detect if the user edits these changes. To do this, we replace the caption with another random caption. In addition, we also provide evaluations of high-quality captions by placing a human reference caption corresponding to the same video. We can then easily verify if the human annotator is doing many edits to low-quality captions while doing a few edits to high-quality captions. The final decision about whether to accept or reject HITs and whether or not the data will be included in the dataset is made by the expert carrying out the validation. An example of the validation flow is depicted in Figure 7.3 and 7.4. The workflow is as follows: first, a list of all the HITs carried out by the workers is shown with red alerts on top when inconsistencies are detected; then the user can inspect and look for details like the time taken to complete the task or the comment given by the annotator; a further inspection can be done clicking in the table and check the response for each caption. In this last popup, the UI shows information such as the original caption, the post-edited caption, if the original caption is correct or has been tampered with, and if the final decision of the validation user is to consider the work done by the annotator as good or not.

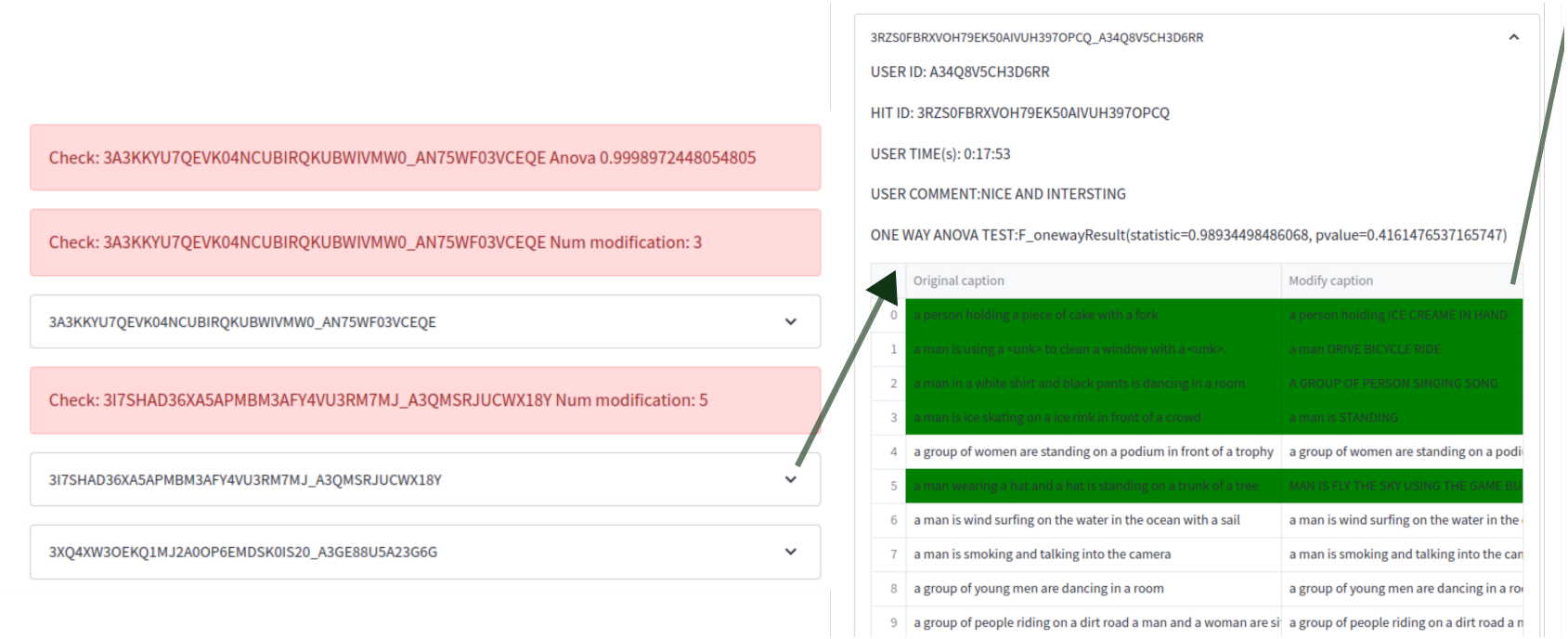
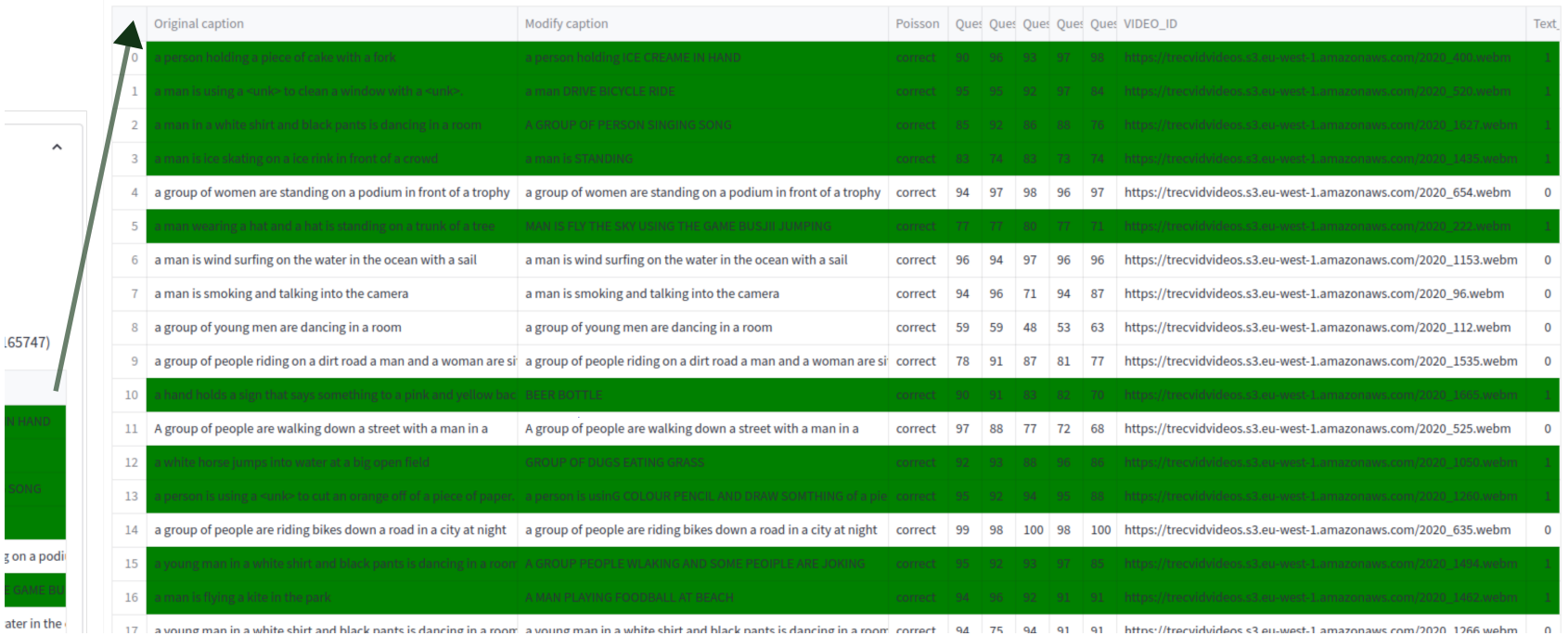


Figure 7.3: Example of the graphic interfaces used for validating the annotations. This is the initial part where we can observe the different alerts and initially view some initial data like the time it took the annotator to complete the task, their comment and some small statistics of their answer.



	Original caption	Modify caption	Poisson	Ques	Ques	Ques	Ques	Ques	VIDEO_ID	Text_
0	a person holding a piece of cake with a fork	a person holding ICE CREAME IN HAND	correct	90	96	93	97	98	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_400.webm	1
1	a man is using a <unk> to clean a window with a <unk>.	a man DRIVE BICYCLE RIDE	correct	95	95	92	97	84	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_520.webm	1
2	a man in a white shirt and black pants is dancing in a room	A GROUP OF PERSON SINGING SONG	correct	85	92	86	88	76	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_1627.webm	1
3	a man is ice skating on a ice rink in front of a crowd	a man is STANDING	correct	83	74	83	73	74	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_1435.webm	1
4	a group of women are standing on a podium in front of a trophy	a group of women are standing on a podium in front of a trophy	correct	94	97	98	96	97	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_654.webm	0
5	a man wearing a hat and a hat is standing on a trunk of a tree	MAN IS FLY THE SKY USING THE GAME BUSJII JUMPING	correct	77	77	80	77	71	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_222.webm	1
6	a man is wind surfing on the water in the ocean with a sail	a man is wind surfing on the water in the ocean with a sail	correct	96	94	97	96	96	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_1153.webm	0
7	a man is smoking and talking into the camera	a man is smoking and talking into the camera	correct	94	96	71	94	87	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_96.webm	0
8	a group of young men are dancing in a room	a group of young men are dancing in a room	correct	59	59	48	53	63	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_112.webm	0
9	a group of people riding on a dirt road a man and a woman are si	a group of people riding on a dirt road a man and a woman are si	correct	78	91	87	81	77	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_1535.webm	0
10	a hand holds a sign that says something to a pink and yellow bac	BEER BOTTLE	correct	90	91	83	82	70	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_1665.webm	1
11	A group of people are walking down a street with a man in a	A group of people are walking down a street with a man in a	correct	97	88	77	72	68	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_525.webm	0
12	a white horse jumps into water at a big open field	GROUP OF DUGS EATING GRASS	correct	92	93	88	96	86	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_1050.webm	1
13	a person is using a <unk> to cut an orange off of a piece of paper.	a person is usinG COLOUR PENCIL AND DRAW SOMTHING of a pie	correct	95	92	94	95	88	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_1260.webm	1
14	a group of people are riding bikes down a road in a city at night	a group of people are riding bikes down a road in a city at night	correct	99	98	100	98	100	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_635.webm	0
15	a young man in a white shirt and black pants is dancing in a room	A GROUP PEOPLE WLAKING AND SOME PEOIPLE ARE JOKING	correct	95	92	93	97	85	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_1494.webm	1
16	a man is flying a kite in the park	A MAN PLAYING FOODBALL AT BEACH	correct	94	96	92	91	91	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_1462.webm	1
17	a young man in a white shirt and black pants is dancing in a room	a young man in a white shirt and black pants is dancing in a room	correct	94	75	94	91	91	https://trecvidvideos.s3.eu-west-1.amazonaws.com/2020_1266.webm	0

Figure 7.4: Example of the graphic interfaces used for validating the annotations. This is the view in more detail here we can see the captions that the user has corrected, highlighted in green, and the scores they gave to the original sentence.

	POES*	Q1	Q2	Q3	Q4	Q5
POES*	1	0.118	0.171	0.180	0.193	0.129
Q1	-	1	0.678	0.701	0.639	0.589
Q2	-	-	1	0.767	0.799	0.625
Q3	-	-	-	1	0.781	0.665
Q4	-	-	-	-	1	0.645
Q5	-	-	-	-	-	1

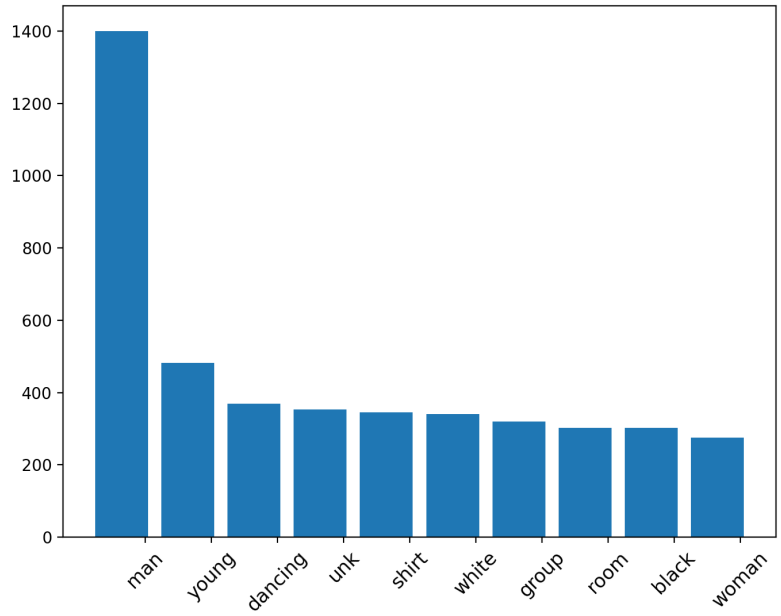
Table 7.1: Kendall's tau correlation between each individual question between each other in TRECVid 2020. Post editing score (POES) is only applied in the captions that have edits while the others are done in the full dataset.

7.4 Results

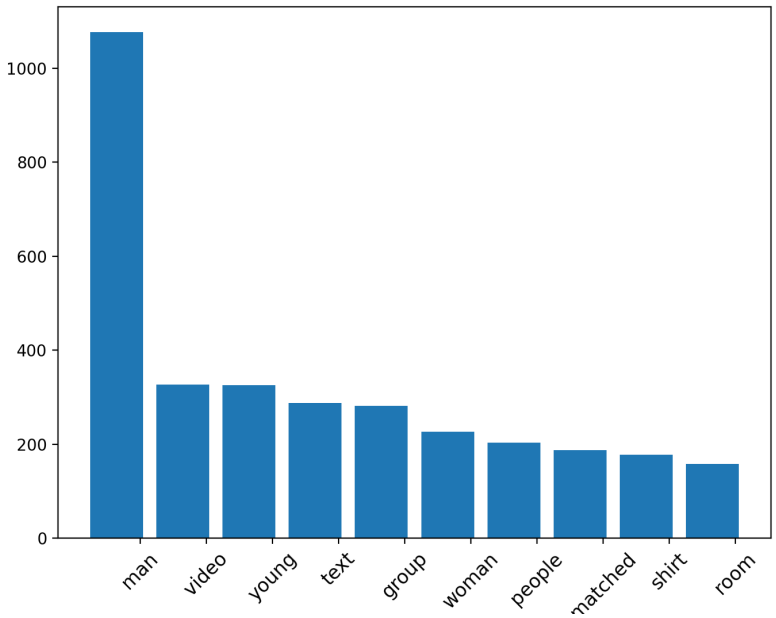
	Q1	Q2	Q3	Q4	Q5
BLEU_4	0.045	0.056	0.061	0.068	0.041
ROUGE	0.051	0.066	0.064	0.074	0.051
METEOR	0.052	0.073	0.077	0.089	0.059
CIDEr	**0.071**	**0.087**	**0.096**	**0.098**	**0.075**
SPICE	0.025	0.016	0.025	0.033	0.024
$BERT_{score}$	0.051	0.066	0.079	0.085	0.057
$CLIP_{score}$	0.060	0.079	0.090	0.096	0.058

Table 7.2: TRECVid-VTT 2019 Kendall's correlation with human assessment. The human judgements are computed as the mean of all the annotators.

In this section, the statistics of the dataset are discussed and some initial results are also presented. The dataset consists of a subset of the captions generated by the systems participating in TreVid 2019 and 2020. TRECVid 2019 [Awad et al., 2020] consists of 10 participating teams and 2020 [Awad et al., 2021] consists of 6 different systems. We use a minimum of 200 randomly selected video-caption pairs per system. The same videos are not used to evaluate each system. At least ten annotators annotate each pair in the final dataset, which is used to evaluate the metrics and obtain final results. However, we also include captions pairs that didn't fulfil these conditions to increase the number of samples in the dataset for other purposes like training newly learned metrics. We use the 2020 dataset for development and 2019 as a test dataset. 2020 contains higher-quality sentences but features fewer captioning systems than 2019.



(a) Histogram of top 10 most used words in the original captions.



(b) Histogram of top 10 most used words in the corrected captions.

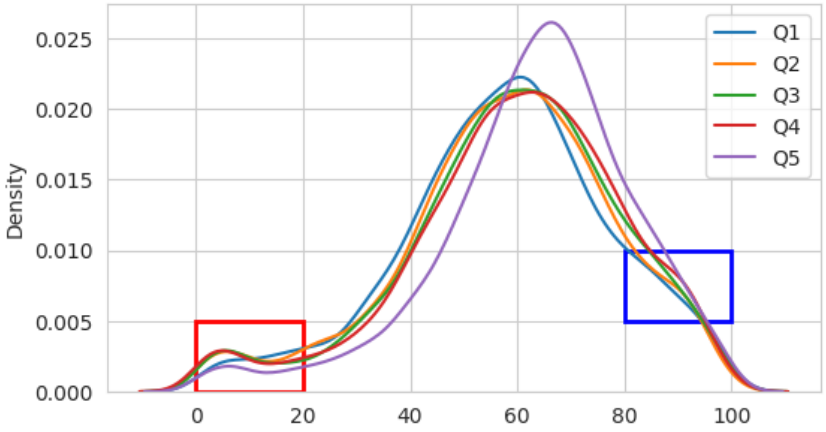
Figure 7.5: Difference in the most used words in the system generated caption compared to the post edited captions.

	Q1	Q2	Q3	Q4	Q5
BLEU_4	0.076	0.094	0.091	0.098	0.080
ROUGE	0.077	0.097	0.094	0.104	0.087
METEOR	0.082	0.101	0.101	0.108	0.094
CIDEr	**0.089**	**0.109**	**0.111**	**0.116**	**0.097**
SPICE	0.025	0.016	0.013	0.013	0.017
$BERT_{score}$	0.067	0.084	0.087	0.091	0.081
$CLIP_{score}$	0.080	0.097	0.102	0.108	0.090

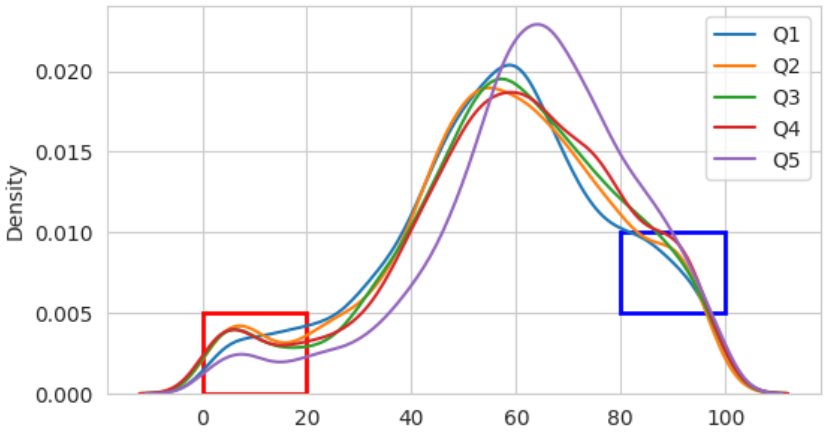
Table 7.3: TRECVid-VTT 2020 Kendall's correlation with human assessment. The human judgments are computed as the mean of all the annotators.

7.4.1 Overall Score Distribution for the Five Evaluation Criteria

First, we analyse the overall score distributions of the five evaluation criteria described above. In the following graph and studies, we use the original score given by the annotator but normalise it so that it can be used to train metrics or applied to any application. This is done to emphasise the original distribution without adding artefacts produced by the normalisation. In Figure 7.6, the kernel distribution of each criterion can be observed. The first distribution includes the artificial quality control sentences added; the second is the distribution without these sentences. In our pipeline, we use two main tampering techniques: replace with a random sentence and replace the original caption with a human reference. These can be observed in the distribution at the start, and the end; and are marked with a square in both figures. The original human sentence can be seen in the part of the blue square and the random ones are part of the red square. As our quality control method involves adding low-quality and high-quality sentences, it can be observed that the distribution is smoother as it contains more caption video pairs falling into these two extremes. One thing to note is that it seems like is that the human-written sentences are also sometimes edited as they can be improved or contain some minor errors. That makes this part of the validation not as truthful as planned and we can see in the graph that their impact on the overall distribution is minor. In a more detailed analysis, the Question with the most distinct behavior in terms of rating



(a) Kernel density plot *with* quality control errors.



(b) Density plot *without* quality control errors.

Figure 7.6: Density plot of TRECVid 2020 human evaluation ratings with and without quality control sentence; squares highlight the main divergences.

distribution is that related to the appearance of the leading person/subject/actor within videos. In most videos, the workers have found that the subject of the video appears in at least one shot. In terms of correlation, table 7.1 shows Kendall's tau correlation between each Question. As expected, the best correlation is between Q2, Q3 and Q4, as they cover similar concepts. Q5 is the most different to the other questions as it is related to what appears in the video, contrary to the others, which are questions related to the quality of the caption and video pair. The post-editing score (POES) is the one with the lowest correlation with the others but it is designed to be used as a metric more than a direct human score. Later we compare it to the commonly used metrics.

Figure 7.1 shows the distribution for the 2019 sub-dataset without the quality control captions. Although it follows a similar distribution we can see that it doesn't include the mentioned gap in the segment of the low-quality captions.

7.4.2 Post-editing Captions to Measure Caption Quality

In the next part, we study the agreement to post edit sentences. As our datasets have multiple annotators per caption, we can identify if all of them agree to edit the caption. 38.42% of the annotators, on average, agree to edit the same caption. To compute this, we take the $\frac{\#\text{times a single caption has been edited}}{\#\text{annotators for this caption}}$. These gave us an idea of the quality of the captions. The higher the number of annotators that agree to edit it, the worst the sentences are. However, we need to consider the post editing as an optional task as they could decide that the caption was acceptable, or some annotators could skip it although the caption was missing parts or was incorrect. For instance, the sentence *a man is playing a ball* was considered correct by most annotators; however, it is missing the word *with*. This means that if a caption is not edited, it does not automatically imply that the quality is sufficient.

Figure 7.7 shows the histograms of similarity between the original caption and the post-edited one. The similarity is computed as:

	Bleu_4	ROUGE	METEOR	CIDEr	SPICE	$CLIP_{text}$
POE 2019	0.450	0.426	0.450	0.747	0.019	0.303
POE 2020	0.279	0.426	0.426	0.784	0.009	0.208

Table 7.4: TRECVid-VTT 2019 and 2020 Kendall's correlation of POES vs all the other metrics.

$$corpus = [\mathbf{oc}, \mathbf{ec}], \tag{7.11}$$

$$vec_o = TF - IDF(\mathbf{oc}, corpus), \tag{7.12}$$

$$vec_e = TF - IDF(\mathbf{ec}, corpus), \tag{7.13}$$

$$Similarity = vec_o * vec_e^T \tag{7.14}$$

where **oc** is the original caption as output by the system and **ec** is the caption edited by the annotator and TF-IDF is the Term Frequency - Inverse Document Frequency. From the histogram, we can observe that most of the captions post-edit are changed into a completely different sentence, and only a small subset are edited with minimum changes. When it was decided to change the description, the whole caption was rewritten. This is likely because the original was of inferior quality or too generic.

Figure 7.5a shows the histogram of the 10 most frequent words in the original caption. Some systems add an additional token *UNK* to mark empty spaces which should include a word. The workers didn't actually see the *UNK* token but saw an extra space in between words. We did this to both avoid confusion and to see if they could actually fill the gap. These kinds of tokens are kept in the final dataset. In comparison, figure 7.5b shows the 10 most frequent words in the modified caption. These graphs are extracted using only the captions that the annotator has fixed. The first thing to notice is that most words change frequency between the two steps, shown as common mistakes produced by all models. One of the main problems that can be extracted from this graph is the overuse of the words man and dancing. Instead, when the model says a person is young, it is usually correct.

7.4.3 Classical Metrics

Tables 7.3 and 7.2 show the Kendall tau correlation of each commonly used metric per Question. In all cases, $CLIP_{score}$, METEOR and CIDEr obtain the best results in all the Questions. These results align with the conclusions of [Shi et al., 2022b] and our own experiments in Chapter 6 which test the metrics in similar datasets.

One main problem when analysing the system-level correlation is that TRECVid 2020 only contains 6 systems and 2019 has only 10. To mitigate this and provide a more realistic result we generated more systems by using the approach in [Graham and Liu, 2016]. In this case, we produce an additional system by combining the captions provided by two other systems, using different percentages to get the maximum number of new systems. The resulting correlation, shown in table 7.5 and 7.6, is a better basis to study how each metric behaves.

	Q1	Q2	Q3	Q4	Q5
BLEU_4	0.822	0.822	0.733	0.777	0.733
ROUGE	0.777	0.777	0.688	0.733	0.688
Meteor	0.822	0.822	0.733	0.777	0.733
CIDEr	0.866	**0.866**	0.777	0.822	0.777
SPICE	0.022	0.111	0.066	0.155	0.199
$BERT_{score}$	0.880	0.762	0.762	0.788	0.774
$CLIP_{score}$	0.733	0.822	0.822	0.866	**0.911**
POES	**0.911**	0.822	**0.911**	**0.955**	**0.911**

Table 7.5: Kendall's tau correlation at system level for TRECVid 2019. 10 real systems were increased to 100 "pseudo-systems".

	Q1	Q2	Q3	Q4	Q5
BLEU_4	0.555	0.591	0.602	0.594	0.597
ROUGE	0.576	0.572	0.570	0.572	0.559
Meteor	0.632	0.670	0.694	0.683	0.644
CIDEr	0.678	0.709	0.723	0.720	0.678
SPICE	0.353	0.374	0.400	0.376	0.321
$BERT_{score}$	0.545	0.600	0.587	0.600	0.590
$CLIP_{score}$	**0.715**	**0.754**	**0.756**	**0.758**	**0.690**
POES	0.706	0.727	0.718	0.723	0.669

Table 7.6: Kendall's tau correlation at the system level for TRECVid 2020. 6 real systems were increased to 100 pseudo-systems.

An interesting result is that these metrics correlated better with the Questions that look for mistakes and about the amount of information in the caption not with Q1 which is the question most similar to the score given in other datasets. This may be connected with the human judgement itself as it was easier to score based on specific errors than one general question as it is the quality of the caption.

7.4.4 Choosing the Best Combination for POES

In table 7.7, we study all the possible combinations of the post-editing score: based on the differences in characters or words and normalised based on the system-generated caption (hypothesis) or the human-edited sentence. This table shows that the best performance compared to the other questions is archived by counting the number of different characters and normalising based on the edited caption.

	Q1	Q2	Q3	Q4	Q5
Character original	0.042	0.088	0.101	0.112	0.066
Character edited	**0.118**	**0.171**	**0.180**	**0.193**	**0.129**
Words original	0.039	0.083	0.096	0.105	0.107
Words edited	0.104	0.155	0.165	0.178	0.117

Table 7.7: Kendall's tau correlation of the post editing scores in TRECVid 2020. It contains the editing based on character and word differences with respect to both the system-generated caption (original) and the edited caption.

Table 7.4 compares the correlation of the new metric POES and all the other metrics. To keep the same domain, $CLIP_text$ only uses the text encoder, so only the system-generated caption and the post-edited caption are used to compute the final score. Interestingly, POES aligns better with CIDEr than the metrics based only on n-grams like BLEU or ROUGE. The likely explanation is that POES also considers the inter-annotation agreement by averaging between the annotators and normalizing based on the length of the fix.

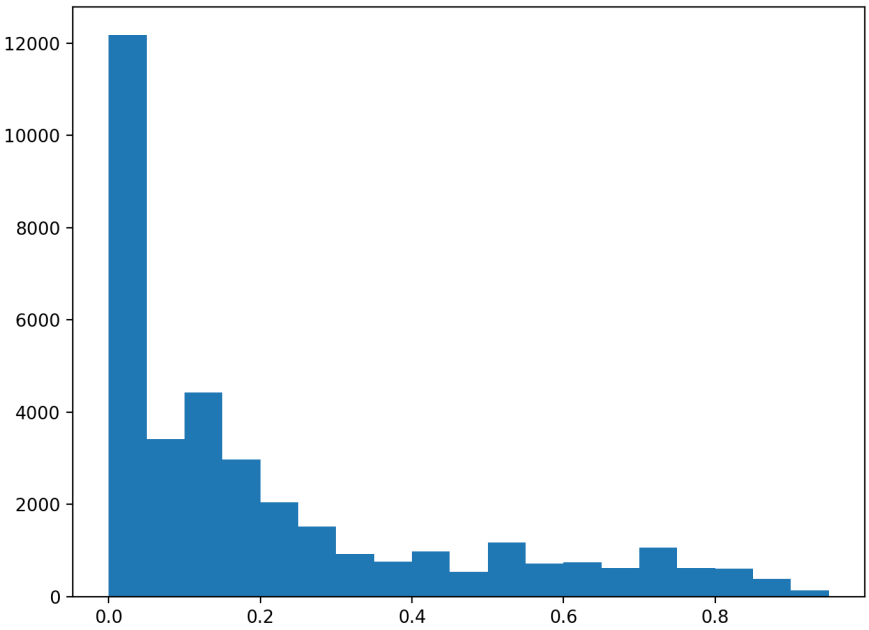


Figure 7.7: Histogram of similarity between sentences that have been modified. Zero means completely different, and one that is the same sentence.

	Original	Correction guided by CLIP	Correction guided by three metrics
METEOR	0.19	0.18	0.20
CIDEr	0.39	0.31	0.36
$CLIP_{score}$	23.9	26.29	25.50

Table 7.8: Study of postediting words replacing them by synonyms and antonyms in the system-generated captions of TRECVid-VTT 2020. The first column refers to the original metrics; the second column shows results using only the CLIP score as a reference to select the sentence; the last column uses the combination of the normalised metrics as a reference.

7.4.5 Automatic Sentences Correction

In previous sections, we saw a change in the distribution of words between the original and the edited sentence. In this experiment, we explore the use of automatic methods to do an initial fix of the sentences without the need for post-edited captions or references.

First, each system caption is divided into words. Then for each noun in the

sentences, all the possible synonyms and antonyms are extracted from Wordnet [Fellbaum, 2010] and a new sentence is generated replacing the original word for each synonym/antonym. Finally, two options for selecting the best sentences are explored: use only $CLIP_{score}$ to select the sentences with the best correlation with the image or use a normalised addition of the metrics with the best correlation with human judgements ($CLIP_{score}$, CIDEr and METEOR).

We can see in table 7.8 that $CLIP_{score}$ always improves using these methods as it is sensitive to synonyms as we only compare the sentences to the images without exploring other transformations. CIDEr on the contrary always decreases doing this however using the three metrics as references provides less drop. METEOR proves that using three references is the best method as it is the only instance that improves. This method helps detect parts of the sentences needing modification in a post-processing step.

7.5 Conclusion

This chapter presents a new dataset for video caption evaluation and a process to collect human judgements. The new dataset is based on direct human assessment and contains five questions to represent relevant concepts in the review of video captions. This reflects 5 different aspects of a caption that are relevant to assessing its quality and completeness. A caption could be acceptable in describing a scene but lacks some central details. Or a caption could include information that is not helpful to represent the subject of the video, which could misdirect the main topic. The aim is to give a numerical value to these concepts using our questions. These could be used to analyse the models and design methods to detect these particularities in the system's outputs.

Additionally, we include the task of post-editing the captions to fix them. With this data, we propose a simple new metric based on the differences in character or words between the original and the post-edit caption. This method is very simple and as a future work, we propose to use a more complete algorithm to compute the

post-editing score like the ones used in the machine translation domain. Also, we have other metrics from the annotators that corrected the original sentences that can be added to the scoring of the captions. For example in an ideal case, the worker would expend more time on captions that are quite similar to the original ones as they need to decide what minimum changes to make to it. Although the new numerical metric provides an excellent reference to determine the quality of the system outputs, the initial edit contains more information that can be exploited to improve the models. For instance, in our study of the frequency of the words between the original caption and the correction, we detect common mistakes that most models make.

We believe that this new data collection process for video captioning could be used to improve the individual systems and the general techniques used across the task. It also provided an excellent baseline to compare them. A deeper analysis could also provide insight into the dataset and how it could be improved.

Finally, we study the automatic correction of the caption in an unsupervised way with word replacement. This is a proof of concept of the idea of doing an edition automatically to improve the quality of the caption. The results prove that the system outputs could be changed to enhance similarity with the human references.

Chapter 8

Conclusions and Further Work

8.1 Thesis Overview

This thesis presents a detailed study of using human judgement to evaluate video captions and how to mimic this procedure with automatic techniques. Metric design is an underdeveloped area of the general field of video captioning, which has seen a rise in interest. The high impact of the references metric selection in the final quality of the output on the models could explain this attraction for research. This conclusion is proven in the study performed in Chapter 3. Two main points are presented related to the choice of metrics: first, even the most popular and complex metrics don't correlate well with human assessment at the system level; second, independent of the architecture or technique used, there is a relationship between the metric used as a reference to determine the performances of the model and the final results in the TRECVid ranking. In this kind of ranking task, the first point is important as not many systems usually participate in this task, so a good correlation is necessary to select the best instance. The second point can be linked to the first, as metrics don't correlate well at the system and caption levels, which is a more difficult task. These affect the performances of the models during training which needs some information on which individual captions are high quality and which are low quality. Commonly these will go through a feedback process which explains why, in general, the references metric greatly influences the final performances of

the model.

Chapter 4 presents an approach to creating a metric that scores more similar to human judgements. Large language models (LLM) have become well known in recent years thanks to their high performances on various NLP tasks. We aim to bring this to the video captioning evaluation model by fine-tuning BERT. We use the direct assessment dataset collected by the organisers of the TRECVid video-to-text task. The amount of labelled data limits this kind of approach, and TRECVid has different quality levels across each workshop run. This motivates the study of Contrastive Learning as presented in Chapter 5. Though we obtained promising results, the diversity in the data provided needs to be improved for such a complex task in order to be truly effective.

Apart from needing more data, another problem is that all the previous evaluation processes only use textual information and are limited by the human references they compare against. As proven in Chapter 4 a pretrained model works well in this task as their knowledge is easy to transfer to another study. BERTHA obtains favourable results, outperforming the commonly used metrics in some setups, answering the question of whether a *metric learned from human evaluations could better correlate with these same human judgements than an engineered metric?*. This also acts as the basis for an initial study for another of the research questions: *Can a learned metric cover all the relevant aspects of a video caption evaluation?*. In Chapter 6, we explore the inclusion of visual information by using CLIP and CLOOB, which are vision and language models (VLM). Similar in design to an LLM, a VLM is trained in a large caption-image pair dataset and is versatile in its applications. Previously they have been proven to work well in image caption evaluation. We study extending this idea to the video domain, where we use multiple frames to refine the evaluation process. A better correlation than the previous metric is then achieved, which can be used even without annotations.

When analysing this dataset in detail, very little can be deduced about the motivation for why scores are given and to what extent the bias of the annotator plays

a role. Also, as shown in the final part of Chapter 6, captions can be inconsistent, and as they are collected from multiple annotators it is challenging to find patterns to understand them. Chapter 7 describes a new data collection pipeline to address these issues while still using the advantage of collecting data from services such as Amazon's Mechanical Turk. The first addition is to divide the qualitative analysis of the caption into five different scores covering some of the main aspects where a caption can fail. The second addition to the process is adding the task of editing the caption to make it correctly describe the video. We use this richer dataset to analyse previously presented metrics and compare them to a new way of evaluating the system, converting the fixes into numerical values.

8.2 Hypothesis and Research Questions Revisited

This section revisits the research questions arising from the overarching research hypothesis presented in Chapter 1 and considers these in light of the various studies conducted and reported in this thesis.

- **RQ1 What changes can be made to a video caption model to improve it in a general way?** In Chapter 3, we present a study that highlights where current models are lacking. By comparing our model to the evolution of each model in the TRECVid workshop, we narrow this to a set of particular challenges faced by models. One of the most common of these is related to the size and variety of videos in the TRECVid-VTT dataset. The use of data augmentation techniques could provide a minimum improvement in general, but to achieve top results, additional datasets are needed to enrich the variety of the videos used for training. However, in this Chapter, we found that the metric used is a significant shortcoming for work in this field. All the most common metrics are based only on measuring the system-generated caption against a single or minimum set of human written sentences. The diversity of these textual descriptions is limited and does not take into account the

motivation of the human annotator when producing that piece of text. Unsurprisingly, compared with the human judgement of the same captions, the correlation results are poor at the system level, which is the simple scenario, and even worse at the caption level. This forms the fundamental motivation for the research conducted in this thesis.

- **RQ2 Can a metric learned from human evaluations better correlate with these same human judgements than an engineered metric?** Both Chapter 4,5 and 6 explore the use of other ways of performing the evaluation. The resulting study has proved that rather than the generally learned metric, fine-tuning metrics trained in a more extensive corpus provided the best correlation with human judgements. This helps address the lack of general data compared to a dataset in a similar task for image captioning or machine translation. These metrics evaluate slightly different aspects of the caption apart from the superficial text similarity.

- **RQ3 Can we find a way to join the visual and textual information to evaluate the sentences used for captioning?** Chapter 6 proves that new developments in vision and language models provided a joint embedding space which can be used to perform this task. These can then include the frame information with minimum modifications in the pipeline. Clustering the frames to select a subset of images provided the best results. This can be explained by the fact that a video is a collection of shots with some transitions, implying that there is both repetitive and noisy information. In some cases, these spaces can also be shifted to adapt better to the task of video captioning evaluation.

- **RQ4 Does a metric cover all the relevant aspects of a video caption evaluation, removing the requirement to report multiple tests or metrics?** This has proven to be a difficult task to explore. For instance, defining all the possible aspects which can influence the study of a caption is

challenging. It is hard to express the concepts in a way that can be integrated into the evaluation pipeline. In Chapter 7, we use the study done previously for natural language generation to define those concepts. We then design a pipeline to collect them. The results are similar to [Vinyals et al., 2016a] in that a good metric will include all of them ito a greater or lesser extent. However, this new dataset provided insights into which parts are lacking and which are more influential in evaluating models. Moreover, evaluating the models using the human workers' post-editing offered additional information to explain the mistakes and a guide to determine what the annotator was looking for in the system-generated sentences. Although we couldn't find a clear answer to this question, the research provided some evidence to design a more robust metric.

8.3 Research Contributions

The contributions of this thesis can be summarised as follows:

- The direct assessment dataset of TRECVid was studied to detect some irregularities. As a result, we developed an updated version that incorporates multiple human judgments for each system-generated caption. Both iterations of the direct assessment dataset are now accessible to the community.

- We propose several methods to train or adapt a model to estimate the evaluation of captions as a human annotator would do. We prove that using a vision and language model archives the best scores even without training. In our experiments, these models achieve top scores across many datasets, and the degree of improvement varies among them.

- Some inconsistencies were detected by the human annotators. From the analysis of these outliers, we propose a new pipeline for direct human assessment. These are linked with the results obtained from a more in-depth analysis of the human judgements where we propose a new collection pipeline in order to

be able to test the metrics in a more detailed way. We collect and release a new dataset following this new pipeline. The most novel part of this dataset is the inclusion of post-edited captions to better comprehend what was lacking from the original caption and how it could be penalised in the final evaluation.

8.4 Proposal for Future Work

Some opportunities and future work that can be investigated as a result of this research are as follows:

- As shown in the thesis, there is a need to have better data to train both the models and to test the evaluation metrics. Although more raw data would be desirable, the study of expanding the current dataset could be interesting. Ideas such as generating "fake" videos from captions could mitigate this problem. This type of approach has proven to work well in other fields like image segmentation.

- Related to the previous point, improvements in self-supervised or semi-supervised techniques could be further explored. Although there is some initial work [Feng et al., 2019], this approach relies on a pre-trained network to provide most of the knowledge. This limits the content of the caption.

- In evaluation, more features could be used to identify more relevant aspects. Audio and motion, in particular, are poorly developed. Speech itself could provide more profound knowledge to generate more valuable captions.

- In this thesis, we perform an initial study of the aspects that determine a caption's quality. Further studies need to be done to improve the evaluation pipeline. This can also help the collection of human judgements tackling specific concepts, and ensuring more homogeneous data.

- We also show some bias and noise in the data. Better techniques to split the two from the available dataset are needed. It would also be interesting if we

could differentiate change in the data coming from the bias of the annotator and which comes from poor annotation.

- The development and training of large vision and language models focused on video data will greatly benefit the field. However, this is hindered by the lack of available data.

8.5 Closing Remarks

In recent years, many models have been integrated into the daily lives of the general public. ChatGPT [Brown et al., 2020a] provides an interface to interact with GPT, which has proven to be an excellent tool for writer assistance and digesting large documents. We see a growing realisation that this model can help improve the simple tasks and routines of many people.

These ideas could also be explored in the video domain where huge datasets of videos can be summarized into short sentences, grouped by content or described in a formal document. It would be interesting to show a long stream to a model and ask for specific descriptions or find specific clips using video caption models.

However, this field is still in its early stages and although some promising results are shown in big challenges like TRECVid-VTT or ACtivitynet, they are not yet ready to be integrated into general applications. Nonetheless, initial work can still be done to integrate them into restricted applications such as suggesting video classification or providing a draft of technical tutorials.

However, it is important to study the implications of using these models, as bias can affect both their performance and output. While reference captions provide a guide for textual descriptions of videos, perfect matching with this set does not necessarily mean perfect results. Therefore, it is crucial to understand the implications of using such information. The author suggests that metrics should be used to explain the criteria and reference back to the video to refine the model for more general or specific descriptions centred around objects and context. In current liter-

ature, testing models seem more like a target to be reached rather than a measure of performance to truly drive progress.

In this thesis, we identify the lack of good evaluation pipelines as a major hindrance to the development of better video caption models. We propose several methods to improve this task, including providing better metrics that correlate more closely with human judgements, and which can be applied without references to captions, and releasing a new dataset with a more detailed analysis of the criteria humans use when evaluating captions. The final note is that we didn't find a general perfect metric that can be used in all the datasets, what we found is some initial methods to evaluate the metrics used and the behaviour of these metrics in specific datasets. The evaluation method we study in this thesis proves to work in a general way but it is also interesting to see how they work in detail as the ones selected for evaluating the final model also influence the final result. In a real scenario, this result will be presented to a final user and that user will expect results "human-like" as is common nowadays with generative models.

Appendix A

Datasets with Human Judgments for Video Captioning

In Table A.1 there is a summary of all the human judgments dataset used in this thesis.

A.1 TRECVid Video To Text Original Evaluation

These datasets refer to the two versions of the TRECVid direct assessment. We use them divided by years to have a more clear split of the data and the participating systems. This dataset is collected using crowdsourcing more specifically Amazon Mechanical Turk. The task that the workers need to do is to evaluate how well the caption describes the video and give a score of 0 to 100. The annotator doesn't have a specific set of rules to score the videos, it doesn't only use their own opinion. The final score is a standardized Z-score per user. To reduce the number of data needed to be annotated, only a subset of the original videos and captions are used

Dataset	Year	Collection process	Videos/ Images	Human judgments (total)
TRECVid-VTT(eval)	2016 - 2020	MTurk	~1500 videos per participating system	1 or 15 annotators
New TRECVid-VTT(eval)	2019 - 2020	MTurk	~3000 videos	~30000
MSCOCO	2015	crowdworkers	2007 images	12
Composite	2015	crowdworkers	4000 images	12000
Flickr8k	2015	crowdworkers	5664 images	17000

Table A.1: Stadistics of the human judgments datasets used in this thesis.

Chapter	Table/Section	SA/MA	VTT'16	VTT'17	VTT'18	VTT'19	VTT'20	
4	Table 4.1	SA	0%	0%	0%	Test 100%	0%	
4	Table 4.2	SA and MA	100%	100%	100%	100%	100%	Test and train
4	Table 4.2	SA and MA	100%	100%	100%	100%	100%	Test and train
4	Table 4.4	SA and MA	100%	100%	100%	100%	100%	Test and train
4	Table 4.5	SA and MA	100%	100%	100%	100%	100%	Test and train
4	Table 4.6	MA	0%	0%	0%	0%	100%	Test and train
4	Table 4.7	MA	100%	100%	100%	100%	100%	Test and train
4	Table 4.8	SA	100%	100%	100%	100%	0%	MT data
4	Table 4.9	SA	0%	0%	100%	0%	0%	Test and train
4	Section 4.4.8	MA	Train 100%	Train 100%	Train 100%	Train 100%	Test 100%	
5	Section 5.4	MA	Train 100%	Train 100%	Train 100%	Train 100%	Test 100%	
5	Table 5.2	MA	Train 100%	Train 100%	Train 100%	Train 100%	0%	Test in MSCOCO
6	Table 6.1	MA	0%	0%	0%	100%	100%	Only middle frame of each video
6	Table 6.2	MA	0%	0%	0%	Test 100%	Test 100%	
6	Table 6.3	MA	Train 100%	Train 100%	Train 100%	Train 100%	Test 100%	
6	Section 6.4.4	MA	0%	0%	0%	0%	100%	Test and train

Table A.2: Summary of the subset of the TRECVid-VTT dataset use in various experiments across the thesis.

in the final evaluation limiting the number of samples at the caption level. Chapter 4 has a more detailed explanation of the reason for having two sets of annotations. Briefly, the original dataset only had one annotator (SA) but that created some discrepancies, that's why we extended the number of annotators per sample to 15 annotators(MA).

Across the chapters, there are mentions of many splits of this dataset. The detail of each experiment can be found in the corresponding Chapter. Table A.2 shows the split use in each experiment across the differnt Chapters.

A.1.1 Machine Translation Data

Chapter 4 also include a few experiments augmenting the data using data from Machine Translation [Ma et al., 2019]. This dataset provides also direct assessment data by comparing system-generated phrases and human-written translations. The collection process is similar to the TRECVid one in terms of the range of scores and postprocessing. Table A.2 points in which experiments this data is used. In the experiments with combine test set, the data is divided in; 70% train, 15% validation and 15% test.

A.2 2015 MSCOCO Image Caption Challenge

This dataset is used in both Chapter 5 and 6. These dataset consist on the system-level evaluations of the 2015 workshop of MSCOCO, share in [Vinyals et al., 2016a]. The human evaluators where ask a series of questions which lead to the following scores:

M1 Percentage of captions evaluated as better or equal to human caption.

M2 Percentage of captions that pass the Turing Test.

M3 Average correctness of the captions on a scale of 1-5 (incorrect – correct).

M4 Average amount of detail of the captions on a scale of 1-5 (lack of details – very detailed).

M5 Percentage of captions that are similar to human description.

The dataset is used in a variety of works in both image caption and video caption evaluation. Commonly only M1 and M2 are used as references for evaluating the performance of the metrics, as they are the only ones that are directly related to the general score to evaluated captions. However, its limited number of samples, only 12 scores, makes it unreliable for modern standards[Hessel et al., 2021].

A.3 Composite

Composite [Aditya et al., 2015] is only used in Chapter 6. It is a dataset which is built from images and captions coming from other datasets: Flickr8k, Flickr30 and MSCOCO. Each image has five human references from which one is reserved to be also rated by humans together with the system-generated captions. In the literature, there are two ways to evaluate this dataset as it has 3 human judgements per caption. The first one is to consider all of them as separate samples and compute the correlation along all the samples. The second way is to average the 3 human

scores per caption and compute the correlation across the average. In this thesis, we follow the second option as suggested in [Hessel et al., 2021].

A.4 Flickr8k-Expert

This dataset is also only used in Chapter 6. It is similar to composite in that it uses images from Flickr8k. However, the annotation process is different. The annotator needs to evaluate the caption on a scale of 1 to 4 (4="caption describes the image without any errors"; 1="caption is unrelated to the image").

A.5 New pipeline TRECVid Video To Text Evaluation

The new dataset collected using the pipeline described in Chapter 7 is also only used in the experiments of that chapter. The videos and system produce captions come from the task as in Chapter 4. The process is the major difference between the two datasets. In this case the annotator is ask to answer five question in a scale of 0 to 100 which refer to specific concepts to evaluate in the comparison between the caption and the original video. The second task they are ask is to modify the caption in order to make it more in line with the content of the video. These alternative tasks originate some distinct behaviour in the correlation with the metrics. More details about this dataset can be found in Chapter 7.

Bibliography

[Aditya et al., 2015] Aditya, S., Yang, Y., Baral, C., Fermuller, C., and Aloimonos, Y. (2015). From images to sentences through scene description graphs using commonsense reasoning and knowledge. *arXiv preprint arXiv:1511.03292*.

[Aggarwal et al., 2020] Aggarwal, A., Chauhan, A., Kumar, D., Mittal, M., Roy, S., and Kim, T.-h. (2020). Video caption based searching using end-to-end dense captioning and sentence embeddings. *Symmetry*, 12(6):992.

[Agrawal et al., 2020] Agrawal, P., Yadav, R., Yadav, V., De, K., and Pratim Roy, P. (2020). Caption-based region extraction in images. In *Proceedings of 3rd International Conference on Computer Vision and Image Processing*, pages 27–38. Springer.

[Al Sharou and Specia, 2022] Al Sharou, K. and Specia, L. (2022). A taxonomy and study of critical errors in machine translation. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 171–179.

[Anderson et al., 2016] Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2016). Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer.

[Awad et al., 2019] Awad, G., Butt, A., Curtis, K., Lee, Y., Fiscus, J., Godil, A., Delgado, A., Smeaton, A. F., Graham, Y., Kraaij, W., and Quénot, G. (2019). Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In *Proceedings of TRECVid 2019*. NIST, USA.

[Awad et al., 2021] Awad, G., Butt, A. A., Curtis, K., Fiscus, J., Godil, A., Lee, Y., Delgado, A., Zhang, J., Godard, E., Chocot, B., et al. (2021). Trecvid 2020: A comprehensive campaign for evaluating video retrieval tasks across multiple application domains. *arXiv preprint arXiv:2104.13473*.

[Awad et al., 2020] Awad, G., Butt, A. A., Curtis, K., Lee, Y., Fiscus, J., Godil, A., Delgado, A., Zhang, J., Godard, E., Diduch, L., et al. (2020). Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. *arXiv preprint arXiv:2009.09984*.

[Awad et al., 2020] Awad, G., Butt, A. A., Curtis, K., Lee, Y., Fiscus, J., Godil, A., Delgado, A., Zhang, J., Godard, E., Diduch, L., Smeaton, A. F., Graham, Y., Kraaij, W., and Quenot, G. (2020). TRECVid 2019: An Evaluation Campaign to Benchmark Video Activity Detection, Video Captioning and Matching, and Video Search & Retrieval. *arXiv e-prints*, page arXiv:2009.09984.

[Awad et al., 2018] Awad, G., Butt, A. A., Curtis, K., Lee, Y., Fiscus, J., Godil, A., Joy, D., Delgado, A., Smeaton, A. F., Graham, Y., Kraaij, W., QuÃ©not, G., Magalhaes, J., Semedo, D., and Blasi, S. (2018). Trecvid 2018: Benchmarking video activity detection, video captioning and matching, video storytelling linking and video search. In *Proceedings of TRECVid 2018*, Gaithersburg, MD.

[Awad et al., 2017] Awad, G., Butt, A. A., Fiscus, J., Joy, D., Delgado, A., Michel, M., Smeaton, A. F., Graham, Y., Kraaij, W., Quenot, G., Eskevich, M., Ordelman, R., Jones, G. J. F., and Huet, B. (2017). Trecvid 2017: Evaluating ad-hoc and instance video search, event detection, video captioning and hyperlinking. In *Proceedings of TRECVid 2017*, Gaithersburg, MD.

[Awad et al., 2023] Awad, G., Curtis, K., Butt, A., Fiscus, J., Godil, A., Lee, Y., Delgado, A., Godard, E., Diduch, L., Liu, J., et al. (2023). An overview on the evaluated video retrieval tasks at trecvid 2022. *arXiv preprint arXiv:2306.13118*.

[Banerjee and Lavie, 2005] Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

[Brown et al., 2020a] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020a). Language models are few-shot learners.

[Brown et al., 2020b] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020b). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

[Caba Heilbron et al., 2015] Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J. (2015). 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.

[Chen and Dolan, 2011] Chen, D. and Dolan, W. B. (2011). Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200.

[Chen et al.,] Chen, S., Song, Y., Zhao, Y., Jin, Q., Zeng, Z., Liu, B., Fu, J., and Hauptmann, A. Activitynet 2019 Task 3: Exploring Contexts for Dense Captioning Events in Videos. Technical report.

[Chen et al., 2020a] Chen, X., Fan, H., Girshick, R., and He, K. (2020a). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

[Chen et al., 2015] Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015). Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

[Chen et al., 2020b] Chen, Y.-C., Li, L., Yu, L., Kholy, A. E., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. (2020b). Uniter: Universal image-text representation learning. In *ECCV*.

[Cheng et al., 2020] Cheng, P., Hao, W., Dai, S., Liu, J., Gan, Z., and Carin, L. (2020). Club: A contrastive log-ratio upper bound of mutual information. In *ICML*, pages 1779–1788. PMLR.

[Chopra et al., 2005] Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

[Cook, 1977] Cook, R. D. (1977). Detection of influential observation in linear regression. *Technometrics*, 19(1):15–18.

[Cui et al., 2018] Cui, Y., Yang, G., Veit, A., Huang, X., and Belongie, S. (2018). Learning to evaluate image captioning. In *CVPR*, pages 5804–5812.

[Das et al., 2013] Das, P., Xu, C., Doell, R. F., and Corso, J. J. (2013). A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2634–2641.

[DeCastellarnau, 2018] DeCastellarnau, A. (2018). A classification of response scale characteristics that affect data quality: a literature review. *Quality & quantity*, 52(4):1523–1559.

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

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, pages 4171–4186.

[Dosovitskiy et al., 2021] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

[Fellbaum, 2010] Fellbaum, C. (2010). Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer.

[Feng et al., 2019] Feng, Y., Ma, L., Liu, W., and Luo, J. (2019). Unsupervised image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4125–4134.

[Fix and Hodges, 1989] Fix, E. and Hodges, J. L. (1989). Discriminatory analysis. nonparametric discrimination: Consistency properties. *Int Stat Rev*, 57(3):238–247.

[Fürst et al., 2021] Fürst, A., Rumetshofer, E., Tran, V., Ramsauer, H., Tang, F., Lehner, J., Kreil, D., Kopp, M., Klambauer, G., Bitto-Nemling, A., et al. (2021). Cloob: Modern hopfield networks with infoloob outperform clip. *arXiv preprint arXiv:2110.11316*.

[Goyal et al., 2017] Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

[Graham, 2015] Graham, Y. (2015). Re-evaluating automatic summarization with bleu and 192 shades of rouge. In *Proceedings of the 2015 Conference on Empir-*

ical Methods in Natural Language Processing, pages 128–137, Lisbon, Portugal. Association for Computational Linguistics.

[Graham et al., 2018] Graham, Y., Awad, G., and Smeaton, A. (2018). Evaluation of automatic video captioning using direct assessment. *PLOS ONE*, 13(9):1–20.

[Graham and Liu, 2016] Graham, Y. and Liu, Q. (2016). Achieving accurate conclusions in evaluation of automatic machine translation metrics. Association for Computational Linguistics.

[Graham et al., 2015] Graham, Y., Mathur, N., and Baldwin, T. (2015). Accurate evaluation of segment-level machine translation metrics. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, Denver, Colorado.

[Green et al., 2013] Green, S., Heer, J., and Manning, C. D. (2013). The efficacy of human post-editing for language translation. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 439–448.

[Guo et al., 2016] Guo, Z., Gao, L., Song, J., Xu, X., Shao, J., and Shen, H. T. (2016). Attention-based lstm with semantic consistency for videos captioning. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 357–361.

[Gutmann and Hyvärinen, 2010] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

[Hanckmann et al., 2012] Hanckmann, P., Schutte, K., and Burghouts, G. J. (2012). Automated textual descriptions for a wide range of video events with 48 human actions. In *European Conference on Computer Vision*, pages 372–380. Springer.

[Hara et al., 2017] Hara, K., Kataoka, H., and Satoh, Y. (2017). Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 3154–3160.

[Hessel et al., 2021] Hessel, J., Holtzman, A., Forbes, M., Bras, R. L., and Choi, Y. (2021). CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hodosh et al., 2013] Hodosh, M., Young, P., and Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *IJCAI*, 47:853–899.

[Hori et al., 2017] Hori, C., Hori, T., Lee, T.-Y., Zhang, Z., Harsham, B., Hershey, J. R., Marks, T. K., and Sumi, K. (2017). Attention-based multimodal fusion for video description. In *Proceedings of the IEEE international conference on computer vision*, pages 4193–4202.

[Howcroft et al.,] Howcroft, D., Belz, A., Clinciu, M., Gkatzia, D., Hasan, S., Mahamood, S., Mille, S., and van Miltenburg, E. Twenty years of confusion in human evaluation.

[Howcroft et al., 2020] Howcroft, D. M., Belz, A., Clinciu, M.-A., Gkatzia, D., Hasan, S. A., Mahamood, S., Mille, S., Van Miltenburg, E., Santhanam, S., and Rieser, V. (2020). Twenty years of confusion in human evaluation: Nlg needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182.

[Huang et al., 2020] Huang, Y., Chen, J., Ouyang, W., Wan, W., and Xue, Y. (2020). Image captioning with end-to-end attribute detection and subsequent attributes prediction. *IEEE Transactions on Image processing*, 29:4013–4026.

[Iashin and Rahtu, 2020a] Iashin, V. and Rahtu, E. (2020a). A better use of audio-visual cues: Dense video captioning with bi-modal transformer. *arXiv preprint arXiv:2005.08271*.

[Iashin and Rahtu, 2020b] Iashin, V. and Rahtu, E. (2020b). Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 958–959.

[Jiang et al., 2020] Jiang, D., Li, W., Cao, M., Zou, W., and Li, X. (2020). Speech simclr: Combining contrastive and reconstruction objective for self-supervised speech representation learning. *arXiv preprint arXiv:2010.13991*.

[Jiang et al., 2019] Jiang, M., Huang, Q., Zhang, L., Wang, X., Zhang, P., Gan, Z., Diesner, J., and Gao, J. (2019). Tiger: Text-to-image grounding for image caption evaluation. pages 2141–2152.

[Kingma and Ba, 2014] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

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

[Koller et al., 1991] Koller, D., Heinze, N., and Nagel, H.-H. (1991). Algorithmic characterization of vehicle trajectories from image sequences by motion verbs. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 90–91. IEEE Computer Society.

[Koponen, 2016] Koponen, M. (2016). Is machine translation post-editing worth the effort? a survey of research into post-editing and effort. *The Journal of Specialised Translation*, 25(2):131–148.

[Krishna et al., 2017] Krishna, R., Hata, K., Ren, F., Fei-Fei, L., and Niebles, J. C. (2017). Dense-captioning events in videos. In *International Conference on Computer Vision (ICCV)*.

[Krishnamoorthy et al., 2013] Krishnamoorthy, N., Malkarnenkar, G., Mooney, R., Saenko, K., and Guadarrama, S. (2013). Generating natural-language video descriptions using text-mined knowledge. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

[Lebron et al., 2022] Lebron, L., Graham, Y., McGuinness, K., Kouramas, K., and O'Connor, N. E. (2022). Bertha: Video captioning evaluation via transfer-learned human assessment. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1566–1575, Marseille, France. European Language Resources Association.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[Levenshtein et al., 1966] Levenshtein, V. I. et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

[Li et al., 2016] Li, Y., Song, Y., Cao, L., Tetreault, J., Goldberg, L., Jaimes, A., and Luo, J. (2016). Tgif: A new dataset and benchmark on animated gif description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4641–4650.

[Li et al.,] Li, Y., Yao, T., Pan, Y., Chao, H., and Mei, T. Jointly Localizing and Describing Events for Dense Video Captioning *. Technical report.

[Li et al., 2024] Li, Z., Haroutunian, L., Tumuluri, R., Cohen, P., and Haffari, G. (2024). Improving cross-domain low-resource text generation through llm post-editing: A programmer-interpreter approach. *arXiv preprint arXiv:2402.04609*.

[Lin, 2004] Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2014). Microsoft coco: Common objects in context.

[Lloyd, 1982] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Trans Inf Theory*, 28(2):129–137.

[Lu et al., 2019] Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Adv Neur In*, 32.

[Ma et al., 2019] Ma, Q., Wei, J., Bojar, O., and Graham, Y. (2019). Results of the wmt19 metrics shared task: Segment-level and strong mt systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

[Manning et al., 2014] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

[Mantecón et al., 2019] Mantecón, H. L., Laaksonen, J., Francis, D., and Huet, B. (2019). Picsom and eurecom experiments in trecvid 2019. In *TRECVid*.

[Mathur et al., 2020] Mathur, N., Wei, J., Freitag, M., Ma, Q., and Bojar, O. (2020). Results of the wmt20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 688–725.

[Michie et al., 1995] Michie, D., Spiegelhalter, D. J., Taylor, C. C., and Campbell, J. (1995). *Machine learning, neural and statistical classification*. Ellis Horwood.

[Minsky and Papert, 1969] Minsky, M. and Papert, S. (1969). An introduction to computational geometry. *Cambridge tiass., HIT*, 479(480):104.

[Misra and Maaten, 2020] Misra, I. and Maaten, L. v. d. (2020). Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6707–6717.

[Moctezuma et al., 2022] Moctezuma, D., Ramírez-delReal, T., Ruiz, G., and González-Chávez, O. (2022). Video captioning: a comparative review of where we are and which could be the route.

[Oord et al., 2018] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[Pan et al., 2016] Pan, Y., Mei, T., Yao, T., Li, H., and Rui, Y. (2016). Jointly modeling embedding and translation to bridge video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4594–4602.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

[Pendurkar et al., 2020] Pendurkar, S., Kolpekwar, S., Dhoot, S., Haribhakta, Y. V., and Banerjee, B. (2020). Attention based multi-modal fusion architecture for open-ended video question answering systems. *Procedia Computer Science*, 171:446–455.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Perez-Martin et al., 2021] Perez-Martin, J., Bustos, B., and Pérez, J. (2021). Attentive visual semantic specialized network for video captioning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5767–5774. IEEE.

[Peris et al., 2016] Peris, Á., Bolaños, M., Radeva, P., and Casacuberta, F. (2016). Video description using bidirectional recurrent neural networks. In *International Conference on Artificial Neural Networks*, pages 3–11. Springer.

[Plummer et al., 2015] Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2015). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2641–2649.

[Radford et al., 2021a] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021a). Learning transferable visual models from natural language supervision. In *ICML*.

[Radford et al., 2021b] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021b). Learning transferable visual models from natural language supervision. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

[Radford et al., 2018] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

[Rohrbach et al., 2015] Rohrbach, A., Rohrbach, M., Tandon, N., and Schiele, B. (2015). A dataset for movie description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3202–3212.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation, parallel dis-

tributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71:599–607.

[Schroff et al., 2015] Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

[Sellam et al., 2020] Sellam, T., Das, D., and Parikh, A. P. (2020). Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

[Shen et al., 2021] Shen, S., Li, L. H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.-W., Yao, Z., and Keutzer, K. (2021). How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*.

[Shi et al., 2022a] Shi, Y., Yang, X., Xu, H., Yuan, C., Li, B., Hu, W., and Zha, Z. (2022a). Emscore: Evaluating video captioning via coarse-grained and fine-grained embedding matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*.

[Shi et al., 2022b] Shi, Y., Yang, X., Xu, H., Yuan, C., Li, B., Hu, W., and Zha, Z.-J. (2022b). Emscore: Evaluating video captioning via coarse-grained and fine-grained embedding matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17929–17938.

[Shimanaka et al., 2019] Shimanaka, H., Kajiwara, T., and Komachi, M. (2019). Machine translation evaluation with bert regressor. *arXiv preprint arXiv:1907.12679*.

[Smeaton et al., 2006] Smeaton, A. F., Over, P., and Kraaij, W. (2006). Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA. ACM Press.

[Sohn, 2016] Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

[St et al., 1989] St, L., Wold, S., et al. (1989). Analysis of variance (anova). *Chemometrics and intelligent laboratory systems*, 6(4):259–272.

[Suin and Rajagopalan, 2020] Suin, M. and Rajagopalan, A. (2020). An efficient framework for dense video captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12039–12046.

[Sun and Nevatia, 2014] Sun, C. and Nevatia, R. (2014). Semantic aware video transcription using random forest classifiers. In *European Conference on Computer Vision*, pages 772–786. Springer.

[Sutskever et al., 2014] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

[Thomason et al., 2014] Thomason, J., Venugopalan, S., Guadarrama, S., Saenko, K., and Mooney, R. (2014). Integrating language and vision to generate natural language descriptions of videos in the wild. Technical report, University of Texas at Austin Austin United States.

[Torabi et al., 2015] Torabi, A., Pal, C., Larochelle, H., and Courville, A. (2015). Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*.

[Tran et al., 2014] Tran, D., Bourdev, L. D., Fergus, R., Torresani, L., and Paluri, M. (2014). C3D: generic features for video analysis. *CoRR*, abs/1412.0767.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

[Vedantam et al., 2015] Vedantam, R., Lawrence Zitnick, C., and Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575.

[Vinyals et al., 2016a] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2016a). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663.

[Vinyals et al., 2016b] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2016b). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663.

[Wang et al., 2020] Wang, S., Lan, L., Zhang, X., Dong, G., and Luo, Z. (2020). Object-aware semantics of attention for image captioning. *Multimedia Tools and Applications*, 79(3):2013–2030.

[Wei and Zou, 2019] Wei, J. and Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

[Welleck et al., 2022] Welleck, S., Lu, X., West, P., Brahman, F., Shen, T., Khashabi, D., and Choi, Y. (2022). Generating sequences by learning to self-correct. *arXiv preprint arXiv:2211.00053*.

[Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[Xia et al., 2021] Xia, Q., Huang, H., Duan, N., Zhang, D., Ji, L., Sui, Z., Cui, E., Bharti, T., and Zhou, M. (2021). Xgpt: Cross-modal generative pre-training for image captioning. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 786–797. Springer.

[Xu et al., 2016] Xu, J., Mei, T., Yao, T., and Rui, Y. (2016). Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

[Xu et al., 2015] Xu, R., Xiong, C., Chen, W., and Corso, J. (2015). Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

[Xu et al., 2019] Xu, Y., Xu, D., Hong, X., Ouyang, W., Ji, R., Xu, M., and Zhao, G. (2019). Structured modeling of joint deep feature and prediction refinement for salient object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3789–3798.

[Yamazaki et al., a] Yamazaki, K., Truong, S., Vo, K., Kidd, M., Rainwater, C., Luu, K., and Le, N. Vlcap: Vision-language with contrastive learning for coherent video paragraph captioning.

[Yamazaki et al., b] Yamazaki, K., Vo, K., Truong, S., Raj, B., and Le, N. Vltint: Visual-linguistic transformer-in-transformer for coherent video paragraph captioning.

[Yamazaki et al., 2022] Yamazaki, K., Vo, K., Truong, S., Raj, B., and Le, N. (2022). Vltint: Visual-linguistic transformer-in-transformer for coherent video paragraph captioning. *arXiv e-prints*, page arXiv:2211.15103.

[Yao et al., 2015] Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., and Courville, A. (2015). Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515.

[Yi et al., 2020] Yi, Y., Deng, H., and Hu, J. (2020). Improving image captioning evaluation by considering inter references variance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 985–994.

[Yu et al., 2022] Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. (2022). Coca: Contrastive captioners are image-text foundation models.

[Zhang and Tian, 2016] Zhang, C. and Tian, Y. (2016). Automatic video description generation via lstm with joint two-stream encoding. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2924–2929. IEEE.

[Zhang* et al., 2020] Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

[Zhang et al., 2020] Zhang, Z., Shi, Y., Yuan, C., Li, B., Wang, P., Hu, W., and Zha, Z.-J. (2020). Object relational graph with teacher-recommended learning for video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13278–13288.

[Zheng et al., 2020a] Zheng, Q., Wang, C., and Tao, D. (2020a). Syntax-aware action targeting for video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13096–13105.

[Zheng et al., 2020b] Zheng, Q., Wang, C., and Tao, D. (2020b). Syntax-aware action targeting for video captioning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Zhou et al., 2019] Zhou, L., Kalantidis, Y., Chen, X., Corso, J. J., and Rohrbach, M. (2019). Grounded video description. In *CVPR*.

[Zhu and Yang, 2020] Zhu, L. and Yang, Y. (2020). Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8746–8755.