Video Analysis for Understanding Human Actions and Interactions

Cristian Manuel Rodriguez Opazo

A thesis submitted for the degree of
Doctor of Philosophy at
The Australian National University

September 2021

I, Cristian Rodriguez-Opazo, hereby declare that this thesis titled: "Video Analysis for Understanding Human Actions and Interactions" and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Cristian Manuel Rodriguez Opazo
Research School of Computer Science
The Australian National University
16 September 2021

To my beloved wife and daughter without whom I would never have finished this PhD thesis.

Acknowledgments

What a journey this has been with many people to be grateful. I want to start with my supervisory panel. Professor Hongdong Li, thank you for this incredible opportunity. You have inspired me in many ways. I am in a lack of words to express all the gratitude for your scientific advice and emotional support. You taught me how to do research and handle the long-time period of frustrations. Your dedication to all your students encourages me to be a better professional.

I would also like to thank the rest of my supervisory panel for their support throughout my Ph.D. I have been fortunate to have Professor Stephen Gould onboard. Thank you for your patient, generosity, dedication, and time. You have taught me how to think out-of-the-box, understand the importance of visualization, and interpret results. I also want to thank Dr. Basura Fernando for all his help, support, and all the insightful discussions we had in the early stage of my Ph.D. In short I cannot imagine a better panel, thank you for teaching me so much and helping me grow during my study.

Dr. Edison Marrese-Taylor, I genuinely appreciate all your help. It is good to meet again after a long time. It has been a pleasure to work with you, thank you for reading all my drafts, be patient with my writing, and teach me how to improve it. I hope we can keep collaborating for a long time.

I have had a wonderful time at ANU, and I credit this to the great colleagues with whom I have the opportunity to work with. David, Fatemeh, Sadegh, Dylan, Juan Adarve, Fred, Liu Liu, Yizhak, Dongxu, Xin, Mina, Rodrigo Santa-Cruz, Yonhon Ng, and many more. Especially Yicong, thank you for the great discussions and brainstorming ideas in the last part of my Ph.D. I hope to keep collaborating in the future with new and challenging vision and language tasks. I have learned a lot from you guys, and I hope that I had contributed to you as well.

To the admin team at RSISE, especially Carol Taylor, Rachel Sinnott, and Marie-Claire Milicevic, thank you for were always happy to advise as much as possible.

A special thank you goes to all of the friends I have made during my time in Canberra. Especially Matthew Callaghan, Macarena Rojas, Alvaro Flores, Carolina Salgado, Roberto Benavente, Aldo Mura, and many others. I have good memories of times shared with you all, your friendship and support I really appreciate over the last four years. I also want to thank the old friends that are always around Matias Valdenegro, Pedro Castañeda, Amanda Luchsinger, Daniela Toro, Ljubomir Sikic, Rodrigo Aravena, Paloma Arluciaga, Felipe Santibañez, and many others for being the best friends ever.

The biggest thank you goes to my family. To my beloved wife, for her support in all my endeavors. You give me the strength and happiness to continue my dreams. I cannot imagine having made it this far without her beautiful and unconditional smile and support. To my parents, Hector and Susana, thank you for always supporting me in everything, even if it meant moving to the other side of the world from you. Thank you for your endless effort, encouragement, kindness, and understanding. Without you, I would never have made it to where I am now. To my siblings, Catalina and Roberto, you inspire me in so many ways; this is mine but also yours.

I would also express my gratitude to the financial support made by CONICYT[1] and by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016). Thank you for the stipend and travel fundings, which enable me to live and attend good conferences around the world. With the help of these institutions, I can pursue and accomplish a dream.

[1] BECAS CHILE, Doctorado Becas Chile/2016 - 72160300

Abstract

Each time that we act, our actions are not just conditioned by the spatial information, e.g., objects, people, and the scene where we are involved. These actions are also conditioned temporally with the previous actions that we have done. Indeed, we live in an evolving and dynamic world. To understand what a person is doing, we reason jointly over spatial and temporal information. Intelligent systems that interact with people and perform useful tasks will also require this ability. In light of this need, video analysis has become, in recent years, an essential field in computer vision, providing to the community a wide range of tasks to solve.

In this thesis, we make several contributions to the literature of video analysis, exploring different tasks that aim to understand human actions and interactions. We begin by considering the challenging problem of human action anticipation. In this task, we seek to predict a person's action as early as possible before it is completed. This task is critical for applications where machines have to react to human actions. We introduce a novel approach that forecasts the most plausible future human motion by hallucinating motion representations.

Then, we address the challenging problem of temporal moment localization. It consists of finding the temporal localization of a natural-language query in a long untrimmed video. Although the queries could be anything that is happening within the video, the vast majority of them describe human actions. In contrast with the propose and rank approaches, where methods create or use predefined clips as candidates, we introduce a proposal-free approach that localizes the query by looking at the whole video at once. We also consider the temporal annotations' subjectivity and propose a soft-labelling using a categorical distribution centred on the annotated start and end.

Equipped with a proposal-free architecture, we tackle the temporal moment localization introducing a spatial-temporal graph. We found that one of the limitations of the existing methods is the lack of spatial cues involved in the video and the query, i.e., objects and people. We create six semantically meaningful nodes. Three that are feed with visual features of people, objects, and activities, and the other three that capture the relationship at the language level of the "subject-object," "subject-verb," and "verb-object." We use a language-conditional message-passing algorithm to capture the relationship between nodes and create an improved representation of the activity. A temporal graph uses this new representation to determine the start and end of the query.

Last, we study the problem of fine-grained opinion mining in video review using a multi-modal setting. There is increasing use of video as a source of information for guidance in the shopping process. People use video reviews as a guide to answering what, why, and where to buy something. We tackle this problem using the three

different modalities inherently present in a video —audio, frames, and transcripts— to determine the most relevant aspect of the product under review and the sentiment polarity of the reviewer upon that aspect. We propose an early fusion mechanism of the three modalities. In this approach, we fuse the three different modalities at the sentence level. It is a general framework that does not lay in any strict constraints on the individual encodings of the audio, video frames and transcripts.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last few centuries, researchers from different fields have devoted significant efforts to understanding human behaviours. Starting from philosophy asking the essential questions of where we go and what we are, psychology and sociology have described and classified behaviour aiming to help people in their daily life. In the last decades, engineers and computer scientists have also put great effort into understanding human behaviour principally for the old dream of creating a robot that might work collaboratively with humans in their daily life. To accomplish such a task, we need to transfer many abilities and knowledge to computers. In light of this, Computer Vision Researchers had made significant progress in the task of face detection, gesture recognition, sentiment analysis, action recognition, action detection, action anticipation, among others. However, we are far behind the capabilities of human vision.

Early works on understanding human behaviour are done in controlled and constrained images or videos. They mostly rely on hand-crafted features and algorithms that describe the spatial relationship of points, and add temporal reasoning if it is needed —usually failing to achieve applicability in unconstrained and uncontrolled environments. Nowadays, with the increased amount of data and computing resources, the most significant progress in computer vision is made by fully data-driven approaches [LeCun et al., 1989; Bishop et al., 1995]. Deep learning methods opened a new era of computer vision outperforming classic methods by a large margin in fundamental tasks like object recognition and detection in images. Such methods encourage researchers to work on more challenging and unconstrained benchmarks.

Undoubtedly, videos have become part of our daily life. There is a clear trend in recent years to record any event, as well as consume videos. Recent YouTube statistics[1] in 2020 show that there are 2 billion of users worldwide generating and consuming video content, and more than 1 billion hours are watched daily, with up to 500+ hours of video uploaded to YouTube every minute.

Surveillance videos are another good example; metropolitan cities around the world are constantly monitored by thousands of cameras located in strategic points to safeguard and help citizens. Australia has doubled the amount of CCTV cameras in the last decade with more than 1 million. The city of Sydney alone has 12.35

[1] https://au.oberlo.com/blog/youtube-statistics

Hugging Handshake Shoot Basketball

Figure 1.1: Different samples of human actions using still images. Actions hugging, handshake and shooting basketball can be seen in the left, middle and right image respectively.(Images extracted from Wikipedia.)

cameras for every 1000 people [2]. However, Australia is far from being the country with more surveillance, as China has installed close to 200 million CCTV cameras across the country which amounts to approximately one camera per seven citizens. [3]

Analysing these data usually requires costly manual annotations that describe a general impression of what the video is about and rarely have fine-grained information of what is happening inside an interval of the video. Therefore, web search engines, like YouTube, commonly rely on textual data such as description or tags to retrieve relevant videos, which make the process inefficient. The analysis of surveillance videos is even more laborious. They are usually long untrimmed videos, clueless of what it is happening on them, and without the help of machine learning algorithms that automatically process the information it is not clear if CCTV cameras are helping to reduce crime [4].

In this thesis, we tackle different problems to understand human actions in videos. First, we introduce an approach to solving action anticipation by hallucinating video representation. Then, we propose two different approaches to solve the temporal localisation of a query in a video. Finally, we present a multi-modal method for mining opinions of video reviews.

1.1 Action Anticipation

When interacting with other people, humans have the ability to anticipate the behaviour of others and act accordingly. This ability comes naturally to us, and we make use of it subconsciously. Almost all human interactions rely on this *action-anticipation* capability. For example, when we greet each other, we tend to anticipate what is the most likely response and act slightly proactively. When driving a car, an experienced driver can often predict the behaviour of other road users. Tennis players predict the trajectory of the ball by observing the movements of the opponent. The ability to anticipate the action of others is essential for our social life and even

[2] https://www.smh.com.au/national/nsw/sydney-in-the-top-15-cities-for-surveillance-levels-20190820-p52irf.html

[3] https://www.nytimes.com/2018/07/08/business/china-surveillance-technology.html

[4] https://electronics.howstuffworks.com/police-camera-crime1.htm

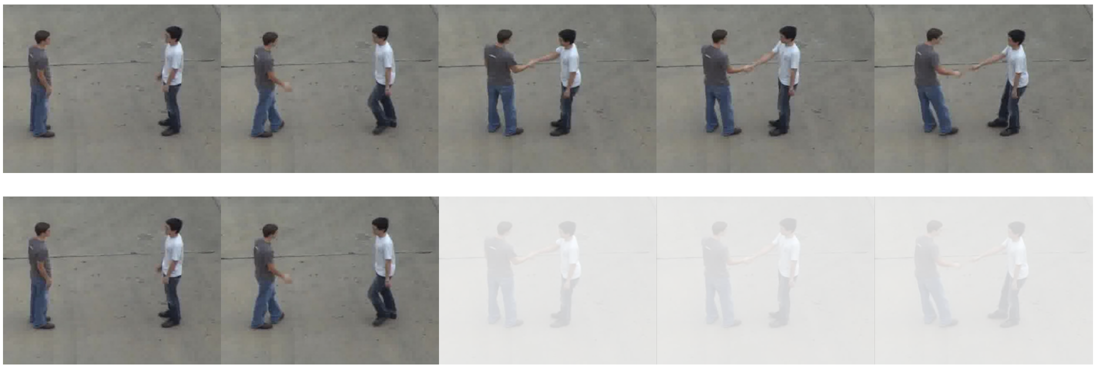


Figure 1.2: An example of the action anticipation task. The first row shows a video sample from the UT-Interaction dataset of the Handshake action. The second row illustrates how much information is used for action anticipation methods. The ambiguity in the first part of the video is distinguishable. The first part of a video can lead to a variety of possible actions.

survival. It is critical to transfer this ability to computers so that we can build smarter robots in the future, with better social interaction abilities that think and act fast. In computer vision, this topic is referred to as *action anticipation* [Ma et al., 2016a; Ryoo, 2011; Aliakbarian et al., 2017; Soomro et al., 2016a,b] or early action prediction.

To contextualize the "action" concept, we use the hierarchical taxonomy defined by Moeslund et al. [2006]: *action primitive*, *action* and *activities*. Consider the following examples: In basketball, “playing basketball” itself can be seen as an *activity*. It involves many *actions*, such as “crossover”‘, “euro step” or “layup”. Each of these actions are the composition of many other *action primitives* such as,“jump”, “run”, “turn-left”, “turn-right” or “dribble”. Another *activity* such as “eating pizza” might involve *actions* including, e.g., “eating”, “cutting” or “chew” and each of those actions can be decomposed in *action primitives*. In summary, an *action primitive* (or movement) describes a basic and atomic motion entity out of which actions are built. *Activities* are a set of several *actions* that typically depend on the context of the environment, objects, or interacting humans.

Although action anticipation is somewhat similar to *action recognition*, they differ by the information which is being exploited. Action-recognition processes the entire action within a video and generates a category label, whereas action-anticipation aims to recognise the action *as early as possible*, even before the entire video is seen. More precisely, action-anticipation needs to predict the future action labels by processing fewer image frames (from the incoming video), even if the human action is still in progress.

In Chapter 3, we present our approach that forecasts motion representation of the seen videos, in specific dynamic images, to anticipate what is the most likely action that is happening in the video.

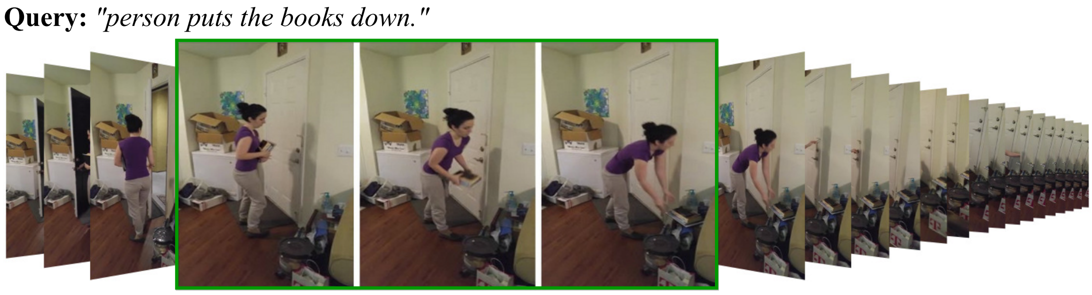


Figure 1.3: An illustration of temporal localization of a natural language query in an untrimmed video. Given a query and a video the task is to identify the temporal start and end of the sentence in the video.

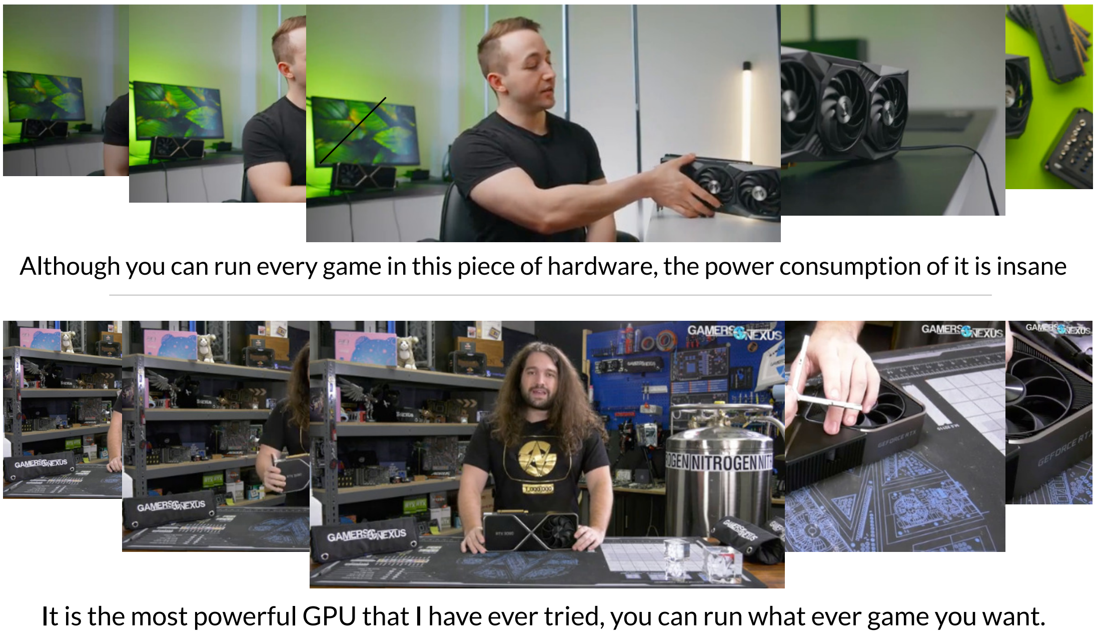
1.2 Temporal Localization of a Query in a Video

Vision-and-language understanding is an important problem that has drawn increasing attention from the computer vision community over the past few years. This research area includes tasks such as video captioning and video question answering. While promising results have been achieved on these tasks, much work still needs to be done to help identify and trim informative video segments in longer videos, while aligning them with relevant textual descriptions. For this reason, tasks such as automatically recognising *when* an activity is happening in a video, have recently become a crucial endeavor in computer vision.

As the amount of video data available to the public continues to grow, searching for particular visual events in large video collections has become increasingly relevant for search engines. This growth has helped draw increased attention to the task of activity localisation in recent years. Activity localisation is an essential and vital task, which has vast applicability — just considering how laborious and error-prone manual annotation can be, even for a small number of videos. In this sense, it is clear that search engines have to retrieve videos not only based on the video metadata but that they also must exploit their internal information in order to accurately localise a given query. Applicability to research areas such as video surveillance, video editing and robotics [Liu et al., 2019], among others, has also helped bring interest to this task.

Earlier works in this context have focused on *temporal action localization* [Richard et al., 2018; Lin et al., 2017; Escorcia et al., 2016a; Chao et al., 2018; Gao et al., 2017c; Xu et al., 2019], which attempts to localize interesting actions in a video from a predefined set of actions. However, this approach constrains the search engine to a relatively small and unrealistic set of (pre-defined) queries from users.

Contemporarily, the task of temporal action localisation with natural language [Gao et al., 2017a; Hendricks et al., 2017] has been proposed to address this issue. Concretely, given a query, the goal is to determine the start and end locations of the associated video segment in a long untrimmed video. About this task, we are specifically interested in the problem of natural language-based queries, or temporal sentence



Although you can run every game in this piece of hardware, the power consumption of it is insane

It is the most powerful GPU that I have ever tried, you can run what ever game you want.

Figure 1.4: Extracted videos from Youtube to illustrate the task of multi-modal opinion mining for video review. In these videos the reviewers are assessing a GPU. They both agree that the GPU can run every game which is a positive comment but one of the reviewers raise the attention in the power consumption of the GPU, which is a negative opinion of a feature or aspect of this GPU.

localisation in the video. Formally, given an untrimmed video and a natural language query, the task is to identify the start and endpoints of the video segment (i.e., moment) that best correspond to the given query, as depicted in Figure 1.3.

In Chapters 4 and 5, we present a proposal-free method and spatio-temporal graph to tackle the temporal moment localisation task.

1.3 Multi-modal Opinion Mining

From the steep growth of video content that is released through the Internet every day, a significant portion is related to video reviews. People explain the most important features or aspects of a product and what are their opinions and sentiments for those aspects. The importance of looking for a second opinion is mainly because on many occasions we need the help of an expert in order to make the right decision. For example, when we want to buy a new cell phone, we usually ask family and friends opinions of the screen, sound, battery and other aspects of the product, and also we investigate what people are saying about the cellphone on the Internet. We can find written reviews of the product which explain the different aspects of it and how good those are compared to other models or brands. We also can search for video reviews that usually show us in a more detailed manner all of these aspects visually through the frames in the videos and the opinion of the expert is said through the video.

The task of opinion mining and sentiment analysis is very well known for the natural language processing community. The goal of the task is identifying and extracting opinions and emotions from the written content. As a result, sentiment can be automatically collected, analysed and summarised for consumers to buy with more information or for companies to improve and create better products. Although reviews often come under the form of a written commentary, people are increasingly turning to video platforms such as YouTube looking for product reviews to help them shop.

Video reviews are naturally different from written reviews given the multi modality inherent in videos, i.e., images, audio and transcript. In this scenario, the modalities complement each other providing rich information of what the reviewer wants to express to the audience. This complementary between modalities means that although the content of each channel may be comprehended in isolation, in theory, we need to process the information in all the channels simultaneously to fully comprehend the message [Hasan et al., 2019]. In this context, information extracted from the nonverbal language in videos, such as gestures and facial expressions, as well as from audio in the manner of voice inflections or pauses, and scenes, objects or images in the video, become critical for performing well.

In this context, Marrese-Taylor et al. [2017] explored a new direction, arguing that video reviews are the natural evolution of written product reviews and introduced a dataset of annotated video product review transcripts. Similarly, Garcia et al. [2019b] recently presented an improved version of the POM movie review dataset [Park et al., 2014], with annotated fine-grained opinions. Although the videos in these kinds of datasets represent a rich multi-modal source of opinions, the features of the language in them may fundamentally differ from written reviews.

In Chapters 6, we present a multi-modal approach to tackle the opinion mining in video reviews.

1.4 Contributions

This thesis contributes to video analysis for understanding human actions and interactions, proposing methods that exploit the inherent components of a video, i.e., images, audio and language. Focusing on the spatial and temporal components of videos. Our main contribution can be described as follows:

1. **A forecasting method of video representation for action anticipation.** We tackle the task of action anticipation by proposing a method that forecasts motion representations. Said generated motion representations are used to predict the action that will happen in the video. This approach relies on dynamic images as a motion representation. It exploits the way that the dynamic images are constructed to propose a set of loss functions that enforce the network to forecast dynamic images that are useful for action anticipation. Using this approach, we achieve a state of the art performance in the action anticipation

task.

2. **A proposal-free method for localising queries in a video.** Proposal and ranking methods for the task of localising queries in a long untrimmed video are computationally expensive, and typically proposals do not deal correctly with the length variability of the actions. Motivated by this problem, we propose to analyse the whole video at once using an attention mechanism. We propose a dynamic filter that is constructed using the natural language query. This filter makes sure that the video reacts to the query in the locations where the action is happening. To enforce this reaction, we propose a loss function that penalises the attention mechanism when attending features outside of the action span. We also consider the problem of subjectivity in the annotation of the start and end of human actions by using soft labels to cope with annotation uncertainties. With these contributions, we could replace the proposal and ranking approach and obtain state of the art performance in the temporal moment localisation task.

3. **Spatio-temporal graph to consider object information in the localisation of a query.** Human actions are often performed in interaction with other humans or objects. To localise a query that describes the human activity in a video, e.g., *"The person with blue shorts is opening the fridge"*, we need to capture the relationships between the objects and humans involved in the scene. Leveraging our proposal-free method, we proposed a spatio-temporal graph that captures the interaction between objects and human entities throughout the video. The graph is conditioned in the natural language query to localise the query in the video properly. After adding the spatial information to our method, we improved the performance of our proposal-free method by a large margin.

4. **A multi-modal approach to fine-grained opinion mining on video reviews.** Video reviews are a significant portion of the video content that is uploaded every day to the Internet. We address the task of opinion mining on video reviews by proposing a multi-modal approach that takes in consideration the visual and audio information, inherent in a video, and determines the aspects of the product in review and the sentiment polarity of the opinion. We proposed an early fusion mechanism that fuses the multi-modal information to classify each of the words in the transcript. Using this approach, we demonstrated that the use of visual and audio information in a typical natural language task is beneficial to obtain better performance.

1.5 Thesis Outline

The remaining chapters of this thesis are summarised below:

Chapter 2 – Background. This chapter provides a general overview of the existing literature related to video analysis and human action understanding, in order to put the contributions of this thesis in context. We also introduced some essential technical background that is assumed in the remainder of the thesis. Finally, we outline various pre-existing datasets that we use for training and evaluating our models, including datasets designed for action recognition and anticipation, temporal moment localisation and multi-modal opinion mining.

Chapter 3 – Action Anticipation by Predicting Future Dynamic Images. In this chapter, we introduce the task of action anticipation. First, we review existing methods on action anticipation from classical methods using hand-crafted features to a new era with deep learning. Later, we present our approach for action anticipation through forecasting dynamic images. Our method is evaluated on standard benchmark datasets showing that quantitatively our approach outperforms prior work. We also present a qualitative analysis of the forecasting video representations.

Chapter 4 – Proposal-Free Temporal Moment Localisation Using Guided Attention. This chapter introduces the relatively new task of localising a natural language query in a long untrimmed video. Then, we review existing methods for resolving this task. We propose a method that removes the proposals step from the current pipelines using a guided attention mechanism. Moreover, we acknowledge the subjectivity in the temporal localisation task. Therefore, we use a soft-labelling approach to reduce the disagreement between annotators. We evaluate our approach in standard benchmarks for the temporal moment localisation task outperforming prior methods.

Chapter 5 – Spatio-Temporal Graph for Temporal Moment Localisation. In this chapter, we present a method that places objects and human relationships at the centre of the temporal localisation of queries in a video through generating a Spatio-temporal graph that captures the relationships between the entities using a language conditioned message-passing scheme. Our method is evaluated in line with the standard benchmark and a new instructional cooking dataset, the method outperforms prior works. Qualitative results are also presented as part of the experiment.

Chapter 6 – Mining Opinions in Video Reviews. In this chapter, We start by describing the fine-grained opinion mining task and motivating the necessity of a multi-modal approach to a classic natural language problem. Then, we develop an early fusion approach that can classify the most relevant aspects that are mentioned in the video and the sentiment of the reviewer about that aspect. We evaluate our method with two benchmarks that we adapt for this task. Our method outperforms standard baselines in the natural language community by just adding the video and audio information.

Chapter 7 – Conclusion and Future Directions. We conclude the thesis with a summary of our main contributions and discussion of future research directions for improving the work.

1.6 List of Publications

Much of the work described in this thesis has been previously published or accepted in conference proceedings, as follows

- Our action anticipation method that forecasts a video representation. (*Published at Anticipating Human Behaviour workshop in ECCV*) [Rodriguez-Opazo et al., 2018].

- The guided attention for a proposal-free method to solve the task of temporal moment localisation. (*Published at WACV*) [Rodriguez-Opazo et al., 2020].

- The Spatio-temporal graph with a language condition message passing for temporal moment localisation (*Published at WACV*) [Rodriguez-Opazo et al., 2021].

- A multi-modal opinion mining approach for aspect extraction and sentiment classification of video reviews. (*Published at Second Grand-Challenge and Workshop on Multimodal Language in ACL*) [Marrese-Taylor* et al., 2020].

The author also contributed to the following projects and publications. This work does not form part of this thesis:

- A word-level sign language recognition dataset and baseline comparisions. (*Published at WACV*) [Li et al., 2020].

- A sub instruction aware approach for vision-and-language navigation. *(Published at EMNLP)*) [Hong* et al., 2020].

- Language-conditioned relational graphs for reasoning in visual-and-language navigation. (*Published at NeurIPS*) [Hong et al., 2020]

- New dataset for understanding human assembly furnitures. This dataset contains actions, segmentation and poses annotations. (*Published at WACV*) [Ben-Shabat et al., 2021]

[*] Equal contribution

Chapter 2

Background

In this chapter, we provide a brief technical introduction of the different theory, architectures and models that have been used throughout this thesis. First, we introduce different ways to represent motion in video clips. Then, we explain Convolutional Neural Networks (CNNs), which we use in this thesis for feature extraction and classification. Following with a brief background of word representations and recurrent neural networks which we extensively use in the temporal moment localisation task. Finally, we briefly contextualise the type of datasets used throughout the thesis to understand better the following chapters in this thesis.

2.1 Motion representation

To understand human actions and interactions, we can naturally decompose the video information in spatial and temporal components [Xu and Li, 2003; Blank et al., 2005; Yilmaz and Shah, 2005; Simonyan and Zisserman, 2014]. The spatial information is carried by individual frames which provide the appearance of the scene and objects. The temporal information can be captured in the form of motion across frames. This motion can help to disambiguate subtle difference between actions that cannot rely on the clues provided by the appearance, e.g., sign language recognition or indoor activities. In this section, we briefly explain two methods to represent motion between frames that have been extensively used for action recognition. We start with the optical flow, which is a fundamental process for motion estimation based on appearance. Then, we explain "dynamic images" which provide a motion representation of consecutive ordered frames, summarising a video clip in a single image. In this thesis, we use dynamic images as a motion representation for the action anticipation task.

2.1.1 Optical Flow

Optical flow or motion estimation is a fundamental problem in computer vision. It consists of computing the apparent motion of objects caused by the relative movement between the objects and camera in consecutive frames of the video sequence.

Assuming that the pixel intensities of an object are constant between consecutive frames, as follows,

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{2.1}$$

where dx and dy represent the motion of the pixel in x and y direction over a time interval dt and, also assuming an small movement between frames, then we can develop the image constraints with Taylor expansion, as follows,

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \epsilon \tag{2.2}$$

where ϵ contains high order terms. Taken the limit of $dt \to 0$ then we can ignore ϵ, which leaves to:

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0 \tag{2.3}$$

Dividing each term by dt we have:

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} = -\frac{\partial I}{\partial t} \tag{2.4}$$

We can recognise some terms in this equation $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ are the first-order derivatives of the image intensity along the axes of the respective image. These can be estimated using *edge detection* algorithms. The partial derivative $\frac{\partial I}{\partial t}$ is the rate of change of image intensity with time and also can be estimated using the difference between frames. The other two terms are related to the *optical flow*. They describe the movement of the pixels along the two axes $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$. Unfortunately, we cannot solve the optical flow components using just this equation since we have two unknowns and only one equation. It is for that reason that a variety of methods introduce additional constraints or assumptions to find another equation and estimate the optical flow [Baker and Matthews, 2004; Brox et al., 2004].

The components of optical flow can be represented as an intensity image, as can be seen in Figure 2.1. Recently, deep learning methods use those motion representation as an input to a neural network to classify actions [Simonyan and Zisserman, 2014; Bilen et al., 2016; Carreira and Zisserman, 2017].

2.1.2 Dynamic Images

Dynamic images, proposed by Bilen et al. [2016, 2017], are a compact motion representation of videos which has shown to be useful for human action recognition. This compact motion representation summarises the temporal evolution of a short video clip (e.g., 10 frames) to a single still RGB image. Dynamic images are constructed using the rank pooling principle [Fernando et al., 2017]. Rank pooling represents a video as a ranking function of its frames $I_1, ..., I_n$, which intention is capture the order of the frames in a video using a ranking of the features.

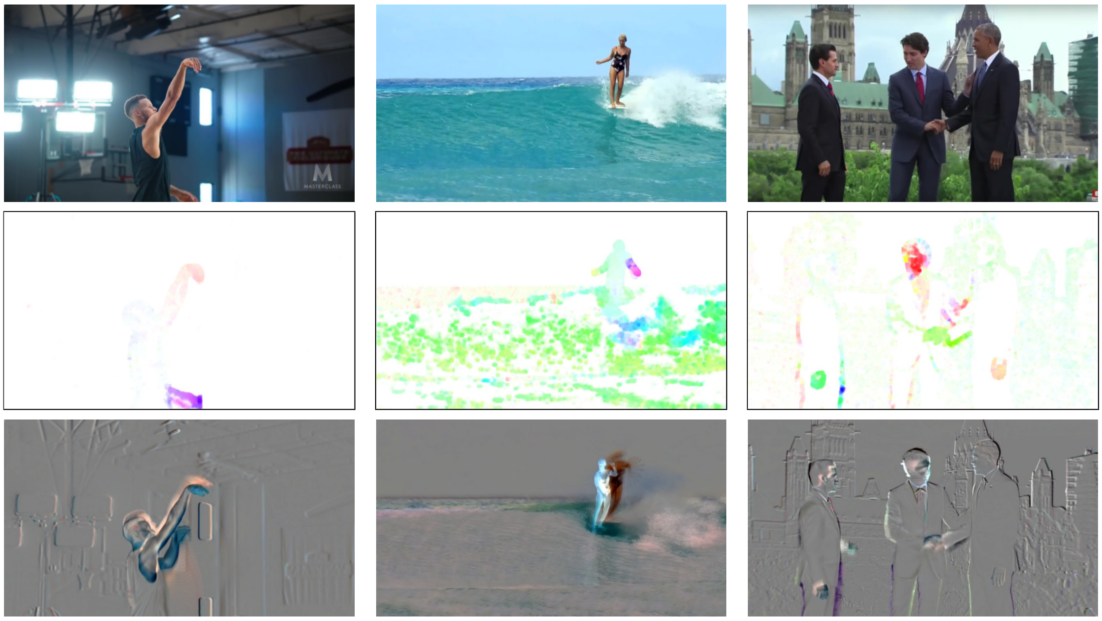


Figure 2.1: Motion representation. First row shows the last frame of a video sequence, second row shows the representation of the motion using optical flow and the last row is the motion representation using dynamic image.

In more details, let $\psi(I_t) \in \mathbb{R}^d$ be a feature vector extracted from the t-th frame in the video and $V_t = \frac{1}{t}\sum_{\tau=1}^{t} \psi(I_\tau)$ be time average of these features up to time t. The ranking function predicts a ranking score for each frame at time t denoted by $S(t|\mathbf{d}) = \langle \mathbf{d}, V_t \rangle$, where $\mathbf{d} \in \mathbb{R}^d$ is a parameter vector of the linear ranking function [Fernando et al., 2017].

The parameter set $\mathbf{d}$ are learned so that the scores reflect the rank of the frames in the video. Therefore, the ranking scores for later times (e.g., q where $q > t$) are associated with larger scores, i.e., $S(q|\mathbf{d}) > S(t|\mathbf{d})$ whenever $q > t$. Learning $\mathbf{d}$ is posed as a convex optimisation problem using the RankSVM [Smola and Schölkopf, 2004] formulation given as the following equation 2.5.

$$\begin{gathered}\mathbf{d}^\star = \rho(I_1, ..., I_t; \psi) = \operatorname*{argmin}_{\mathbf{d}} E(\mathbf{d}), \\ E(\mathbf{d}) = \frac{\lambda}{2}||\mathbf{d}||^2 + \\ \frac{2}{T(T-1)} \sum_{q>t} \max\{0, 1 - S(q|\mathbf{d}) + S(t|\mathbf{d})\}.\end{gathered} \tag{2.5}$$

The first term in this objective function is the usual quadratic regularizer used in SVMs. The second term is a hinge-loss soft-counting how many pairs $q > t$ are incorrectly ranked by the scoring function. Note in particular that a pair is considered correctly ranked only if scores are separated by at least a unit margin, i.e. $S(q|d) > S(t|d) + 1$. Optimising equation 2.5 defines a function $\rho(I_1, \cdot, I_T; \psi)$ that maps a video sequence of length T to a single vector denoted by $\mathbf{d}^\star$. Since this vector contains enough information to rank all frames in the video clip, it can be assumed that it can aggregate temporal information from all frames. Therefore, it can be used as a video motion representation or a temporal representation. The process of constructing $\mathbf{d}^\star$ from a sequence of video frames is the idea behind *rank pooling*.

When one applies rank pooling directly on the RGB image instead of the feature space, the resulting $\mathbf{d}^\star$ is an image known as the *dynamic image*. In this case, $\psi(I_t)$ is an operator that takes RGB images as input. Similarly, $\mathbf{d}^\star$ has the same number of elements as a single frame of the input video. In this case, the dynamic image $\mathbf{d}^\star$ summarises the temporal information from the video sequence. Bilen et al. [2016] presents an approximation to rank pooling, which it is much faster and works as well in practice.

The derivation of the rank pooling approximation is based on the idea of considering the first step in a gradient based optimisation of the equation 2.5. Starting with $\mathbf{d} = \vec{0}$, the first approximated solution obtained by gradient descent is $\mathbf{d}^\star = \vec{0} - \eta \nabla E(\mathbf{d})|_{\mathbf{d}=\vec{0}} \propto -\nabla E(\mathbf{d})|_{\mathbf{d}=\vec{0}}$ for any $\eta > 0$, where

$$\begin{aligned} \nabla E(\vec{0}) &\propto \sum_{q>t} \nabla \max\{0, 1 - S(q|\mathbf{d}) + S(t|\mathbf{d})\}|_{\mathbf{d}=\vec{0}}. \\ &= \sum_{q>t} \nabla \langle \mathbf{d}, V_t - V_q \rangle \\ &= \sum_{q>t} V_t - V_q \end{aligned} \tag{2.6}$$

Bilen et al. [2016] by further expanding $\mathbf{d}^\star$ found an approximation to the rank pooling as follows,

$$\mathbf{d}^\star \propto \sum_{q>t} V_q - V_t = \sum_{q>t} \left[\frac{1}{q} \sum_{i=1}^{q} \psi_i - \frac{1}{t} \sum_{j=1}^{t} \psi_j \right] = \sum_{t=1}^{T} \alpha_t \psi_t \tag{2.7}$$

where the coefficients α_t are given by

$$\alpha_t = 2(T - t + 1) - (T + 1)(H_T - H_{t-1}) \tag{2.8}$$

where $H_t = \sum_{i=1}^{t} 1/t$ is the t-th *harmonic number* and $H_0 = 0$. Hence, the rank pooling operator can be reduced to:

$$\hat{\rho}(I_1, ..., I_T; \psi) = \sum_{t=1}^{T} \alpha_t \psi_t \tag{2.9}$$

In summary, the construction of dynamic images using *approximated rank pooling* is a simple weighted sum of the images with the coefficient α. We construct dynamic images using this approximation by taking a weighted sum of input image sequence where weights are given by predefined coefficients α. In this thesis, we make use of dynamic images in chapter 3 as a motion representation for the action anticipation

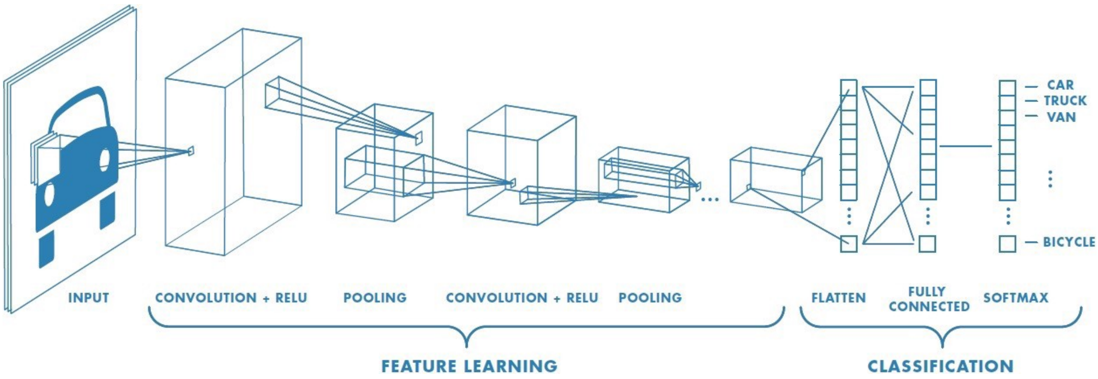


Figure 2.2: An example of the architecture of a Convolutional Neural Networks (Image taken from Stanford University's Introduction to Convolutional Neural Networks course, 2018)

task.

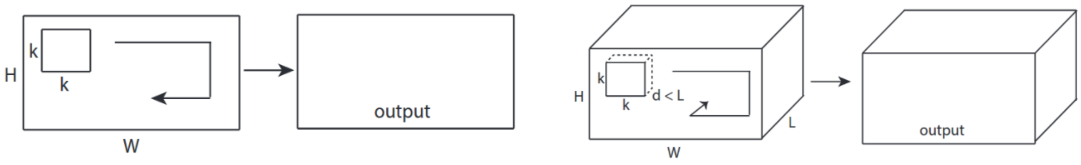
2.2 Convolutional Neural Network

We use 2D convolutional neural networks to label still images or motion representation, such as mentioned in the previous section. Convolutional neural networks (CNNs) are a type of neural network proposed by LeCun et al. [1989] to process gridlike data, such as images. It proposed a convolutional neural network called LeNet-5 which was able to classify hand-written digits from images. Nowadays, CNNs can be seen as a powerful and useful tool to solve many machine learning and computer vision tasks.

A Convolutional Neural Network is constructed using multiple stacks of different layers. Including convolutional, pooling and fully-connected layers, as can be seen in Figure 5. Convolutional layers are computational units that produce an output feature map by convolving an input with linear kernels containing learned parameters. Each kernel operates independently through the input tensor by sliding across the width and height of the image and producing a 2D response map. In contrast with traditional methods that rely on hand-crafted features extraction methods to train classifiers, CNNs can automatically learn a high-dimensional representation of an input image. They use trainable filters, and local neighbourhood pooling operations, which are applied to the input image and sub-sequences feature maps. Convolutions have three properties that are particularly desirable for image processing and feature extraction [Goodfellow et al., 2016].

Equivariance to translation, equivariant functions are ones that if the input changes the output changes in the same way. In the case of images, it means that features can be recognised regardless of their position in the image.

Parameter Sharing refers to using the same parameters for more than one function in the model. In contrast with neural networks, convolutions perform an efficient parameter sharing since the same kernel is used at every position in the input.



(a) (b)

Comparison between a) 2D convolutions and b) 3D convolutions in the way that process the information

Sparse Interaction assuming that the kernel is smaller than the input, only elements that are nearby will interact with the kernel. This means that fewer parameters are needed to be stored, reducing memory requirements.

A Convolutional Neural Network contains multiple stacks of convolutional layers that combine the information between different filters. This multiple stack of layers allows CNNs to learn concepts at an increasing level of abstraction as information is processed through the network. It also usually involves interspersing convolutional layers with non-linear activation function and pooling operations.

Since most of the real-world data is non-linear, Neural Networks require to handle non-linearity. Various non-linear activation functions have been studied since the creation of CNNs such as *tanh* or *sigmoid*. However, the rectified linear unit (ReLU), proposed by Nair and Hinton [2010], is frequently used. ReLu is non-saturating, and it has been found to perform better in most situations. Its output is given by:

$$f(x) = \max(0, x) \tag{2.10}$$

Pooling layer is a form of non-linear down-sampling using a summary statistic such as the mean or maximum value of the incoming vector. It is inserted periodically in-between successive convolutional layers. Pooling has some desirable properties, such as making the network robust to small transformation and distortions since the statistical mean or maximum is the same. It also makes the input representation smaller and more computationally manageable.

The last part of a CNN is a set of fully connected layers, as can bee seen in Figure 2.2. Their purpose is to classify or predict values using the high-level feature representation of the input created by the stack of convolutional and pooling layers. Furthermore, adding a fully connected layer is a way of learning non-linear combination of these features. Fully connected layers expect a 1D real-valued vector as input, which is obtained by flattening the 3D volume of the final pooling layer in the architecture. This vector becomes the input to the fully connected layer which produces non-spatial output. In the classification stage, a softmax function is usually used as the activation function in the output layer of the fully connected layer. The softmax function takes a vector of arbitrary real-valued scores, and it turns to a vector of values between zero and one that sum to one, i.e., in the form of a probability.

An entire Convolutional Neural Network can express a single differentiable function, whose parameters are trained using backpropagation. Back-propagation [Rumelhart et al., 1985], short for "backward propagation of errors", is an algorithm for supervised learning of neural networks that uses gradient descent. The method calculates the gradient of the error function with respect to the neural network's parameters. During training, the parameters are adapted to minimise a given a task-specific loss function. It has been empirically observed many times that CNNs parameters trained for one task can function as an effective visual feature extractor for many other jobs. For example, by simply using the final layer of the convolution layers as a feature representation of the input image. In this thesis, we use CNNs that have been pre-trained for image classification, ImageNet [Russakovsky et al., 2015], and object detection, Visual Genome [Krishna et al., 2016], as generic image encoders.

2.2.1 3D Convolution Neural Networks

Performing action classification in a video seems to be a natural extension of image classification since videos are "just" a sequence of frames. Therefore, a naive approach would be aggregating or fusion all the predictions of every single frame in the video [Karpathy et al., 2014]. However, this approach does not consider the temporal information that is vital to understand the action that a human is performing in a video clip. In light of this, recent approaches that consider the temporal information using 3D convolutions emerge. These approaches are widely inspired by the success of 2D convolutional neural networks in classification of images. In contrast to 2D convolutional neural networks that use filters convolving the image in width and height, 3D convolutions use filters that also convolve the video volume in the depth direction, as can be seen in Figure 2.3

One of the first methods that use 3D CNNs for action recognition is proposed by Tran et al. [2015]. In this work, authors build upon previous 2D CNNs work done by Karpathy et al. [2014]. They repurposed a 3D convolution neural network as a feature extractor. Extensively searching for the best architecture and 3D convolution kernel.

They found that the network focussed on spatial appearance in the first few frames and tracked the motion in the subsequent frames using deconvolutional layers to interpret the decision made by the neural network through a visualisation tool. Later Carreira and Zisserman [2017] proposed an inflated version of a well know 2D convolution neural network that has been used for image classification in the ImageNet challenge.

Although 3D convolution neural networks have shown significant progress for action recognition, there is still work to be done. This method struggles to capture Spatio-temporal information across frames with a subtle difference. A more refined understanding of objects or human motion will allow us to better tackle task like sign-language recognition. Moreover, there is a substantial computational cost. A simple 2D CNN for image classification could use 5 million of parameters compared to six times more used by the same architecture inflated for 3D information. Taking days or months for training this architecture depending on the dataset.

2.3 Word Representations

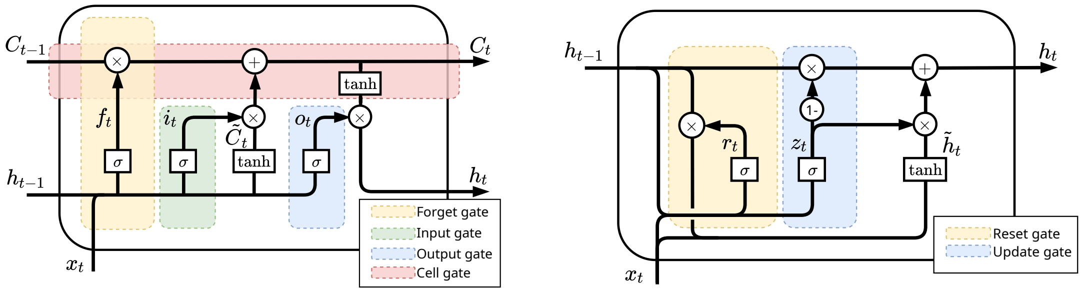
In the intention to combine language and vision to understand the action that a human is performing in a video, it is necessary to create a word representation of the text that is amenable for numerical computation. These representations are usually called embeddings. In the early days of natural language, words were embed using symbolic and fixed representations, such as dictionary lookup and one-hot embeddings. Those methods produce word representations that are easy to make but are hard to train, requiring a large amount of memory, and they do not incorporate meaning into the representation.

Recently methods are designed to consider semantic and syntactic dependencies between words by exploiting the distribution of those in massive datasets. They generate dense real-valued vector in a real space of tens or hundreds of dimensions. Vectors representing similar meaning words like *cookie* and *biscuit* should have small numerical difference since they are closely related.

One of the early used distributed word representations method is Skip-Gram model [Mikolov et al., 2013], also known as Word2Vec. This method can learn distributed word representation in an unsupervised manner. For doing so, the method tries to predict the source context words (surrounding words) given a target word (the centre word). Thus, the words are representing utilising the means of their neighbours.

While Word2Vec captures certain local context window, GloVe exploits overall co-occurrence statistics of words from the corpus, which is a large collection of texts. They construct a matrix of term co-occurrences. For each word, they compute the conditional probability, e.g. for word water $P(k|\text{water})$, where k is a word from vocabulary. If k is the word stream, the value of P would be high. If k is fashion, then the expected value is low as they do not usually co-occur together. As it explained by Pennington et al. [2014], the training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. Owing to the fact that the logarithm of a ratio equals the difference of logarithms, this objective associates ratios of co-occurrence probabilities with vector differences in the word vector space. We extensively use GloVe embeddings in Chapters 4, 5 and 6.

Distributed word representations are powerful techniques. They do not suffer from undesirable properties of simple methods, such as the need for large memory, difficulty to be train and lack of semantic information in the word representation. Although, Distributed word representations can incorporate semantic information of words into their representation. They cannot produce vectors for Out-of-Vocabulary words. Moreover, vector representations for rare words are usually not learned well enough.



Long Short-Term Memory

Gated Recurrent Unit

Figure 2.4: Ilustration of the Recurrent Neural Networks of the Long Short-Term Memory and Gated Recurrent Unit

2.4 Recurrent Neural Networks

Recurrent neural networks (RNNs) are a type of artificial neural networks designed to process sequential data. Unlike conventional feed-forward neural networks that only operate on an input space, RNNs also operate on internal states or memory which trace previous information passed to the network. This structure allows the network to learn sequential dependencies in data. However, the long sequences make it difficult to simple RNNs. Since they suffer to carry the information from earlier inputs in the sequence to later ones.

In the backpropagation step, if a gradient value becomes extremely small, it does not contribute to the learning process. We refer to that issue as the vanishing gradient problem. Usually, earlier layers in an RNN suffer from this issue and forget what is seen in longer sequences, thus having a short-term memory effect.

Long Short-Term Memory Networks and Gated Recurrent Units Networks were created as the solution to short-term memory effect. They have internal mechanisms called gates that can regulate the flow of information. We extensively make use of this type of neural networks to encode the language and video representations within Chapter 4, 5 and 6.

2.4.1 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks, proposed by Hochreiter and Schmidhuber [1997], are an special type of RNN designed to learn long-term dependencies. The LSTM receives an input x_t and the previous hidden state h_{t-1} for each sigle step over the input sequence $\mathbf{x} = \{x_1, x_2, x_3, \cdots, x_t\}$ and output a new hidden state h_t, equation 2.11. In this subsection, we introduce the four gates that compose an LSTM, Figure 2.4a.

$$h_t = LSTM(x_t, h_{t-1}) \tag{2.11}$$

Forget gate decides what information is passed through the cell state. The forget gate consists on a fully connected layer with a sigmoid activation, equation 2.12. It looks at h_{t-1} and x_t and output a number between 0 and 1 for each element in the cell state C_{t-1}.

Input gate decides the new information that will be saved in the cell state. The input gate consists on two parts, a fully connected layer with a sigmoid activation, equation 2.13 that decide which values will be updated, and the candidates new values $\tilde{C}$ which consist on a fully connected layer and a *tanh* activation function, Equation 2.15.

Cell gate updates the old cell state for C_{t-1} with the new cell state C_t. The cell gate uses the forget gate and the input gate to decide what to keep or forget depending on the decisions made in those gates, equation 2.16.

Output gate filters the cell state using a fully connected layer and a sigmoid activation layer to decide what values will be output by the LSTM at time t, Equations 2.14 and 2.17.

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) && (2.12)\\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) && (2.13)\\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) && (2.14)\\
\tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) && (2.15)\\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t && (2.16)\\
h_t &= o_t \odot \tanh(C_t) && (2.17)
\end{aligned}
$$

2.4.2 Gated Recurrent Units Networks

Recently introduced by Cho et al. in 2014 Gated Recurrent Units (GRU) networks is also designed to solve the vanishing gradient problem. The GRU solves the problem similarly to LSTM using gated units. It is composes of two gates, *Update gate* and *Reset gate*, which decide what information passed to the output, Figure 2.4b.

Update gate determines how much information about the previous time steps or past information needs to be passed along to the future, equation 2.19.

Reset gate decides how much of the past information the model needs to forget, equation 2.18.

Although there is no evidence that GRU performs better than LSTM Chung et al. [2014], the resulting model is simpler than standard LSTM models and has been growing increasingly popular.

$$
\begin{aligned}
r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) && (2.18)\\
\tilde{h}_t &= \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h) && (2.19)\\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t && (2.20)
\end{aligned}
$$

2.5 Datasets

In this section, we present the different types of video datasets used throughout this thesis. We follow the categories presented by Klaser [2010] to show that the type of videos that we use throughout the thesis is the most complex. According to Klaser 2010 video datasets can be categories in *controlled*, *constrained* and *unconstrained*.

Controlled videos are acquired using different tools and techniques to facilitate automated processing. For instances, lighting condition can be controlled to better detect human bodies; multiple cameras placed in specific locations to obtain desirable 3D reconstruction. Markers that can be attached to human bodies for detecting joints, limbs and facial gestures. These type of videos are not used throughout the thesis.

Constrained videos are collected with a limited degree of influence in the environment. Therefore, certain assumptions of the environment can be made, e.g., a single person fully visible, favourable light conditions, static cameras or a limited set of expected actions. For instance, videos from surveillance cameras belong to this category since the camera placement and intrinsic parameters are fixed and known. An algorithm such as background subtraction can be used. In this thesis, we made use of three datasets that belong to this category. UT-Interaction [Ryoo and Aggarwal, 2010], TACoS [Rohrbach et al., 2014], and POM [Garcia et al., 2019b]. In the case of UT-Interaction, videos are taken from a surveillance camera and have a limited set of expected actions. Videos in TACoS are taken from a tripod located in front of a kitchen table, and it also has a constrained list of possible actions related to cooking. Finally, POM dataset consists of closeup to human people reviewing movies.

Unconstrained videos are recorded under conditions which cannot be influenced. This is the case for, e.g., TV and cinema-style movie data, sports broadcasts, music videos, or personal amateur clips. Only very few assumptions, if any, of a rather general nature can be made, such as humans are present and relative well visible. The main challenges for this more realistic data include changes of viewpoint, scale, and lighting conditions, partial occlusion of humans and objects, cluttered backgrounds, abrupt movement etc. In this thesis, we made use of seven different unconstrained videos, which are mainly recollected from YouTube. JHMDB [Jhuang et al., 2013], Youtubean [Marrese-Taylor et al., 2017], UCF101 [Soomro et al., 2012], Charades [Sigurdsson et al., 2016], ActivityNet [Caba Heilbron et al., 2015] and YouCookII [Zhou et al., 2018b].

Chapter 3

Action-Anticipation By Predicting Future Dynamic Images

This chapter is based on:

Rodriguez-Opazo, C.; Fernando, B; Li, H. Action Anticipation By Predicting Future Dynamic Images. *Proceedings of the European Conference on Computer Vision*, Workshop on Anticipating Human Behavior, 2018

The human motion in a video is one of the most distinctive clues to understanding human actions. It provides temporal information that can help to recognise and predict the action and interactions of a human. In this chapter, we focus on the human action anticipation task in videos. It is an essential and challenging task. It consists of predicting the action taking place in a trimmed video by observing only a small portion of an action in progress. This task is critical for applications where computers have to react to human actions as early as possible such as in pedestrian intention forecasting system in autonomous vehicles, human-robot interaction, assistive robotics, among others. It is challenging because, with a limited observation of the video, the future action of a human is unknown and highly ambiguous [Piedimonte et al., 2015]. The starting of different actions, such as "walking" and "running", is quite similar.

We propose a method that forecasts the most plausible future human motion by hallucinating a motion representation of a video's small snippet. Instead of directly predicting action labels of a partially seen sequence, we propose a new method that generates future motion representation which will be used in predicting action labels. We represent human motion using dynamic images [Bilen et al., 2016] and use a set of tailored loss functions to encourage and guide the generative model to forecast accurate future motion prediction by exploiting the relationship between static appearance and motion information. Our method yields state-of-the-art performance on the task in different benchmark datasets.

3.1 Introduction

When interacting with other people, human beings have the ability to anticipate the behaviour of others and act accordingly. This ability comes naturally to us, and we make use of it subconsciously. Almost all human interactions rely on this *action-anticipation* capability. For example, when we greet each other, we tend to anticipate what is the most likely response and act slightly proactively. When driving a car, an experienced driver can often predict the behaviour of other road users. Tennis players predict the trajectory of the ball by observing the movements of the opponent. The ability to anticipate the action of others is essential for our social life and even survival. It is critical to transfer this ability to computers so that we can build smarter robots in the future, with better social interaction abilities that think and act fast.

In computer vision, this topic is referred to as *action anticipation* [Ma et al., 2016a; Ryoo, 2011; Aliakbarian et al., 2017; Soomro et al., 2016a,b] or early action prediction. Although action anticipation is to some extent similar to *action recognition*, they differ by the information being exploited. Action-recognition processes the entire action within a video and generates a category label, whereas action-anticipation aims to recognise the action *as early as possible*. More precisely, action-anticipation needs to predict the future action label as early as possible by processing fewer image frames (from the incoming video), even if the human action is still in progress.

Instead of directly predicting action labels [Aliakbarian et al., 2017], we propose a new method that generates future motion representation from partial observations of human action in a video. We argue that the generation of future motion representation is a more intuitive task than generating future appearance, hence easier to achieve. A method that is generating future appearance given the current appearance requires to learn a conditional distribution of factors such as colour, illumination, objects and object parts; therefore, more challenging to achieve. In contrast, a method that learns to predict future motion does not need to learn those factors. Furthermore, motion information is useful for recognising human actions as it is shown by Bilen et al. [2017] and Simonyan and Zisserman [2014] and can be presented in various image forms [Bilen et al., 2017; Ahad et al., 2012].

In this chapter, we propose to predict future motion representation for action anticipation. Our method hallucinates what is in the next motion representation of a video sequence given only a fraction of a video depicting a partial human action. We make use of a convolutional autoencoder network that receives a motion image as input at time t and outputs a motion image for the future (e.g., $t+1$). Using the Markov assumption, we generate more motion images of the future using already generated motion images (i.e., we generate motion images for time $t+1, \cdots, t+k$). Then we process generated motion images using Convolutional Neural Network (CNN) to make action predictions for the future. As we can generate future motion images, now we can predict human actions only observing a few frames of a video containing an action. We train our action anticipation and motion generation network with several loss functions. These loss functions are specifically tailored to generate accurate representations of future motion and to make accurate action predictions.

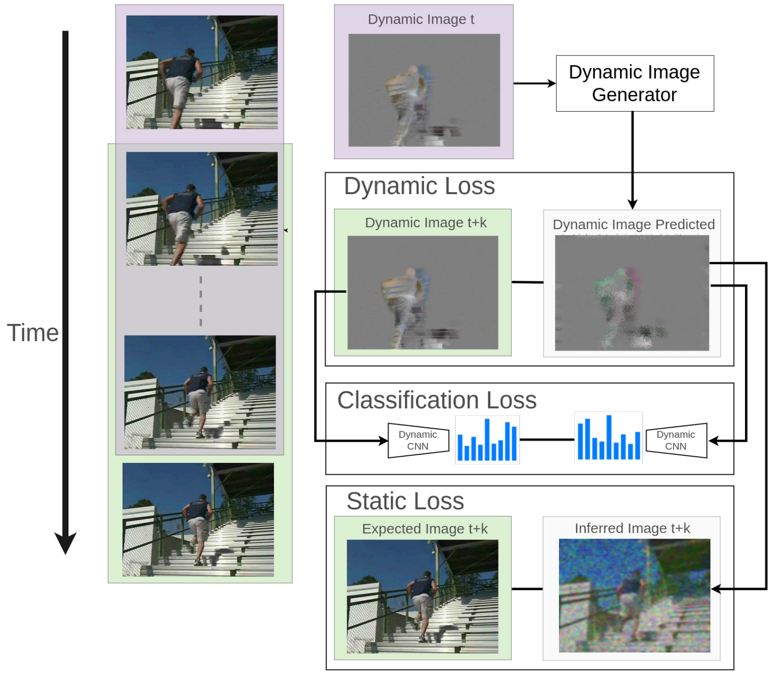


Figure 3.1: Training of our generation module using multiple loss functions. **a)** ***Dynamic Loss*** evaluates the difference in motion information between predicted and ground truth dynamic image using $\mathcal{L}_2$ norm. **b)** ***Classification Loss*** takes care of generating dynamic images that are useful for action anticipation. **c) Static Loss** computes the $\mathcal{L}_2$ norm between predicted and ground truth RGB information at $t + k$ to evaluate the difference in appearance.

Clearly, the motion information depends on the appearance and vice versa. For example, motion representations, such as the optical flow, rely on two consecutive RGB frames. Similarly, the content of dynamic images [Bilen et al., 2017] relies on the appearance of consecutive frames. The relationship between static appearance and motion information is somewhat surprising and mysterious [Carreira and Zisserman, 2017]. Recently, proposed dynamic images have managed to explore this relationship to some degree of success [Bilen et al., 2017]. In particular, dynamic images summarise the temporal evolution of appearance of few frames (e.g., ten frames) into a single image. Therefore, this motion summary image (also known as a dynamic image) captures the motion information of those frames. In this work, we hallucinate dynamic images for the future and use them for the task of action anticipation [1].

We generate dynamic images using both the expected appearance and motion of the future. Specifically, future dynamic images are generated by taking into account both reconstructive loss (coined *dynamic loss*) and future expected appearance, which is coined *static loss*. As motion and appearances should be consistent with each

[1] However, the main concept of this chapter is applicable for other types of motion representation as well, such as optical flow, motion history images, among others.

other, static loss is designed to satisfy expected future appearance in the generated dynamic images. In addition to that our generated dynamic images make use of class information and therefore are discriminative. These loss functions are tailored to generate accurate future dynamic images, as is depicted in Figure 3.1. In summary, in this chapter, we study the motion representation of a video to understand human actions and predict the action in a trimmed video as early as possible by forecasting the motion representation of a small section of the video.

3.2 Related work

Action anticipation literature can be classified into classical methods that use handcrafted features and modern approaches that use a more holistic video representation with deep learning techniques.

In the case of handcrafted features, Ryoo [2011] addressed the problem of early recognition of unfinished activities. Two variants of the bag-of-words representations are introduced to handle the computational issues of modeling how feature distributions change over time. Yu et al. [2012] proposed to use spatial-temporal action matching for the action anticipation task using spatial-temporal implicit shape models. Later, Li and Fu [2014] explore sequence mining where a series of actions and object co-occurrences are encoded as symbolic sequences. Kong et al. [2014] investigate the temporal evolution of human actions to predict the type of action as early as possible. Their approach captures the temporal dynamics of human actions by explicitly considering all the history of observed features as well as features in smaller temporal segments. More recently, Soomro et al. [2016b] propose to use binary SVMs to localise and classify video snippets into sub-action categories and obtain the final class label in an online manner using dynamic programming. Given the need to train one classifier per sub-action, Soomro et al. [2016a] extended this approach using a structural SVM formulation. In addition to that, this method introduces a new objective function to encourage the score of the correct action to increase as time progresses.

While all the above methods utilise handcrafted features, most modern methods use deep learning approaches for action anticipation [Ma et al., 2016a; Aliakbarian et al., 2017; Jain et al., 2016a; Vondrick et al., 2016]. Deep learning-based methods can be primarily categorised into two types; on one side, we have different approaches that rely on novel loss functions for action anticipation [Ma et al., 2016a; Aliakbarian et al., 2017; Jain et al., 2016a]. On the other side, we have methods that try to generate future content by content prediction [Vondrick et al., 2016]. In this context, Ma et al. [2016a] propose to use a Long Short-Term Memory (LSTM) with a ranking loss to model the activity progression and use it for effective action anticipation tasks. They use Convolutional Neural Network (CNN) features and an LSTM to model both spatial and temporal information.

Similarly, Jain et al. [2016a] proposed a new loss function known as the exponentially growing loss. In this work, they try to penalise errors increasingly over time using an LSTM-based framework. In the same line, Aliakbarian et al. [2017] proposed a novel loss function for action anticipation that encourages the network to achieve good predictions as early as possible. The method of Aliakbarian et al. [2017] tries to overcome ambiguities in the early stages of actions by preventing false negatives from the beginning of the sequence. Singh et al. [2017] proposes a slightly different but related approach. Their method works in a different setup, online action detection, that can be used for action anticipation. Instead of predicting the future class label, in Vondrick et al. [2016], the authors propose to predict future visual representation. However, the primary motivation in Vondrick et al. [2016] is to learn activity representations using unlabeled videos.

Our work is different from the work of Vondrick et al. [2016] as we predict future motion representation using dynamic images. We make use reconstruction loss, class information loss, and expected future appearance to guide our neural network to predict the next motion representation images. As our generated dynamic images are trained for action anticipation, they are class-specific and different from original dynamic images [Bilen et al., 2017]. As demonstrated, our generated dynamic images are more effective than original dynamic images for the action anticipation task.

In a similar direction, Gao et al. [2017b] proposed to generate future appearance images using an LSTM autoencoder that uses regression and classification losses to anticipate actions. We argue that predicting future appearance representation is a complicated task, and also we believe that action anticipation can benefit more from motion prediction than the challenging appearance prediction. Predicting future content has been explored on other related problems in different domains of computer vision. Some of the work focuses on predicting (or forecasting) the future trajectories of pedestrians Kitani et al. [2012] or predicting motion from still images Pellegrini et al. [2009]. However, we are the first to show the effectiveness of predicting good motion representations for early action anticipation.

3.3 Method

Our work's objective is to recognise human actions as early as possible from a video sequence depicting human action. We present a method that hallucinates future motion from a partially observed human action sequence (RGB video clip). Then we process these hallucinated future motion representations to make future action predictions, also known as action anticipation. Our motion representation is based on dynamic images [Bilen et al., 2016, 2017]. Dynamic images model dynamic information of a short video clip and summarise motion information to a single frame, more information about dynamic images can be found in the background section 2.1.2 of this thesis. We present a method to hallucinate future dynamic images using a convolutional autoencoder neural network. We process generated dynamic images to predict future human actions using a CNN named *dynamic CNN*. To improve action recognition performance further, we use observed still image appearance information and process them with a *static CNN*. Furthermore, we make use of dynamic

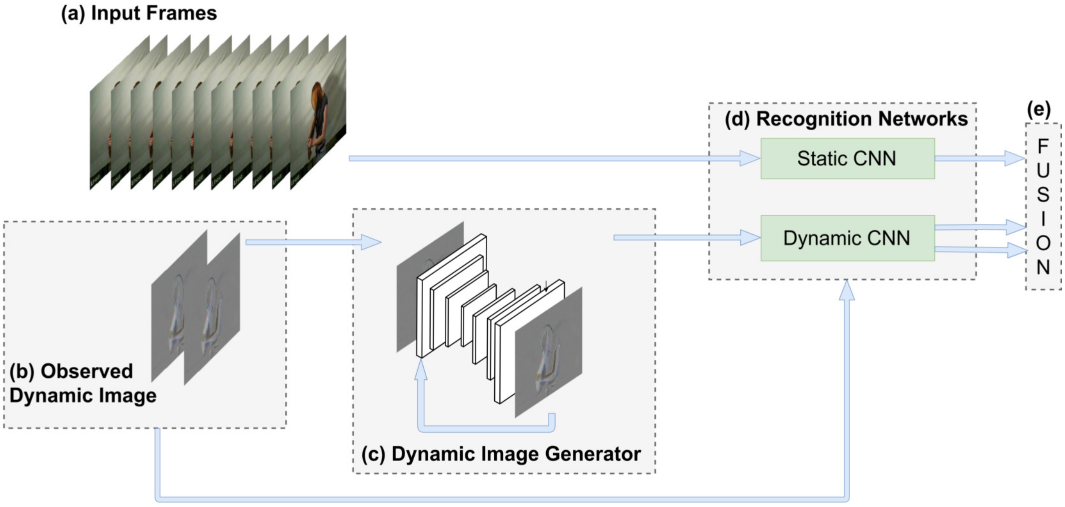


Figure 3.2: **Overview of our anticipation approach**. We receive as an input a sequence of RGB video frames **(a)**. Then we use RGB images with windows size T to compute the Dynamic Images for seen part of the video **(b)**. The last dynamic image of the seen part is used to feed our dynamic image generator and generate $\hat{D}_{t+1}$ **(c)**. Next, we feed *Dynamic* CNN with observed dynamic images and generated dynamic images and *Static* CNN with RGB images **(d)**. Finally, we fusion all the outputs of our recognition networks **(e)**.

images created from observed RGB data and use the same dynamic CNN to make predictions. Therefore, we make use of three kinds of predictions and fuse them to make the final prediction (see Figure 3.2). In the following section, we present our dynamic image generation model in section 3.3.1. Then we discuss loss functions in section 3.3.2 and how to train our model in section 3.3.3.

3.3.1 Future motion prediction

Given a collection of videos X with corresponding human action class labels Y, our aim is to predict the human action label after observing as few frames from the start of each video as possible.

Each video $X_i \in X$ is a sequence of frames $X_i = \langle I_1, I_2, \cdots, I_n \rangle$ of variable length n. We process each sequence of RGB frames to obtain a sequence of dynamic images using equation 2.7. Instead of summarising the entire video with a single dynamic image, we propose to generate multiple dynamic images from a single video sequence using a fixed window size of length T. Therefore, each dynamic image is created using T consecutive frames. We process each training video X_i and obtain a sequence of dynamic images $\langle D_1, D_2, \cdots, D_n \rangle$.

Our objective is to train a model that is able to predict the future dynamic image D_{t+k} given the current dynamic images up to time t i.e., $\langle D_1, D_2, \cdots, D_t \rangle$. Therefore, we aim to model the following conditional probability distribution using a parametric model

$$P(D_{t+k}|\left\langle D_1, D_2, \cdots, D_t \right\rangle; \Theta) \tag{3.1}$$

where Θ are the parameters of our generative model and $k \geq 1$. We simplify this probabilistic model using the Markov assumption on the sequence of dynamic images, hence now $k = 1$ and condition only on the previous dynamic image D_t. Then our model simplifies to following equation 3.2. Note that even though the Markov assumption is applied to the dynamic image, each dynamic image summarises several video frames.

$$P(D_{t+1}|D_t; \Theta) \tag{3.2}$$

The model in equation 3.2 simplifies the training process. Moreover, it may be possible to take advantage of different kinds of machine learning technique to implement the model in equation 3.2.

Now the challenge is to find a good machine learning technique and loss function to train such a model. We use a denoising convolutional autoencoder to hallucinate future dynamic images given the current ones. Our convolutional autoencoder receives a dynamic image at time t and outputs a dynamic image for next time step $t + 1$. In practice, dynamic images up to time t are observed, and we recursively generate dynamic images for time $t + 1, \cdots, t + k$ using the Markov assumption. Although we use a denoising convolutional autoencoder, our idea can also be implemented with other generative models such as autoencoders [Baldi, 2012], variational conditional autoencoders [Kingma et al., 2014; Sohn et al., 2015] and conditional generative adversarial networks [Mirza and Osindero, 2014] The autoencoder we use has four convolution stages. Each convolution has kernels of size 5 × 5 with a stride of 2, and the number of features maps for the convolution layers are set to 64, 128, 256, and 512 respectively. Then the deconvolution is the inverted mirror of the encoding network (see Figure 3.2), which is inspired by the architecture used in DCGAN [Radford et al., 2016]. Next, we discuss suitable loss functions for training the autoencoder.

3.3.2 Loss functions for training the autoencoder

First, we propose to make use of reconstructive loss $\mathcal{L}_2$ distance between predicted dynamic image $\hat{D}_{t+1}$ and the ground truth dynamic image obtained from the training data D_{t+1} as shown in equation 3.3. We called this loss *Dynamic loss* to avoid confusion with other losses that we propose later on that also use $\mathcal{L}_2$.

$$\mathcal{L}_{DL} = ||\hat{D}_{t+1} - D_{t+1}||_2 \tag{3.3}$$

Even though this loss function helps us to generate expected future dynamic image, it does not guarantee that the generated dynamic image is discriminative for action anticipation. Indeed, we would like to generate a dynamic image that contains more discriminative information that helps to perform the action anticipation task.

Therefore, we propose to explore the teacher-student networks [Hinton et al., 2015] to teach the autoencoder to produce dynamic images that would be useful for action anticipation. First, we train a teacher CNN which takes dynamic images as input and produces the action category label. Let us denote this teacher CNN by $f(D_i; \Theta_{cnn})$ where it takes dynamic image D_i and produces the corresponding class label vector $\hat{y}_i$. This teacher CNN that takes dynamic images as input and outputs labels is called *Dynamic CNN* (see Fig 3.2). This teacher CNN is trained with cross-entropy loss [Szegedy et al., 2017]. Let us denote our generator network as $g(D_t; \Theta) \rightarrow D_{t+1}$. We want to take advantage of the teacher network $f(; \Theta_{cnn})$ to guide the student generator $g(D_t; \Theta)$ to produce future dynamic images that are useful for classification. Given a collection of current and future dynamic images with labels, we train the generator with the cross-entropy loss as follows:

$$\mathcal{L}_{CL} = -\sum_t y_i \log f(g(D_t; \Theta); \Theta_{cnn}) \tag{3.4}$$

where we fix the CNN parameter Θ_{cnn}. Obviously, we make the assumption that CNN $f(D_i; \Theta_{cnn})$ is well trained and has good generalization capacity. We ensure this by evaluating the performance of the Dynamic CNN to classify the action label for real dynamic images. We call this loss as the *classification loss* which is denoted by $\mathcal{L}_{CL}$. In theory, compared to original dynamic images [Bilen et al., 2016, 2017], our generated dynamic images are class-specific and therefore discriminative.

Motion and appearance are related. Optical flow depends on the appearance of two consecutive frames. Dynamic images depend on the evolution of appearance of several consecutive frames. Therefore, it is important to verify that generated future motion actually adhere to future expected appearance. Another advantage of using dynamic images to generate future motion is the ability to exploit this property explicitly. We make use of future expected appearance to guide the generator network to produce accurate dynamic images. Let us explain what we mean by this. When we generate future dynamic image D_{t+1}, as demonstrated in equation 3.5, implicitly we also recover the future RGB frame I_{t+1}. Using this equation 3.5, we propose so-called *static loss* (SL) (equation 3.6) that consists of computing the $\mathcal{L}_2$ loss between the generated RGB image $\hat{I}_{t+1}$ and real expected image I_{t+1}.

$$\begin{aligned} D_{t+1} &= \sum_{i=1}^{T} \alpha_i I_{t+1+i} \\ &= \alpha_T I_{T+t+1} \sum_{i=1}^{T-1} \alpha_i I_{t+1+i} \\ I_{T+t+1} &= \frac{D_{t+1} - \sum_{i=1}^{T-1} \alpha_i I_{t+1+i}}{\alpha_T} \end{aligned} \tag{3.5}$$

The applicability of static loss does not limit only to matching the future expected appearance, but also we guide the autoencoder model $g(; \Theta)$ to use all implicitly generated RGB frames from $\hat{I}_{t+2}$ to $\hat{I}_{T+t+1}$ making the future dynamic image better by modelling the evolution of appearance of static images. Indeed, this is a better loss function than simply taking the dynamic loss as in equation 3.3.

$$\mathcal{L}_{SL} = ||\hat{I}_{T+t+1} - I_{T+t+1}||_2 \tag{3.6}$$

3.3.3 Multitask learning

We train our autoencoder with multiple losses, the static loss ($\mathcal{L}_{SL}$), the dynamic loss ($\mathcal{L}_{DL}$) and the classification loss ($\mathcal{L}_{CL}$). By doing so, we aim to generate dynamic images that are good for classification, as well as a representative of future motion. With the intention to enforce all these requirements, we propose to train our autoencoder in a batch-wise multitask manner. Overall, one might write down the global loss function $\mathcal{L} = \lambda_{sl}\mathcal{L}_{SL} + \lambda_{dl}\mathcal{L}_{DL} + \lambda_{cl}\mathcal{L}_{CL}$. However, instead of finding good scalar weights $\lambda_{sl}, \lambda_{dl}$, and λ_{cl}, we propose to divide each batch into three sub-batches and optimise each loss using only one of those sub batches. Therefore, during each batch, we optimise all losses with different sets of data. We found this strategy leads to better generalisation than optimising a linear combination of losses.

3.3.4 Inference

During inference, we receive RGB frames from a video sequence as input. Using those RGB frames, we compute *dynamic images* following equation 2.7 with a window size $T = 10$. In the case that the amount of frames is less than what is needed to compute the dynamic image, we compute the dynamic image with the available frames according to equation 2.7.

We use the last dynamic image (D_t) to predict the following dynamic image ($\hat{D}_{t+1}$). We repeat this process to generate k number of future dynamic images. We process each observed RGB frame I, observed dynamic images D and generated dynamic images $\hat{D}$ by respective static and dynamic CNNs that are trained to make predictions (see Figure 3.2). Then, we obtain a score vector for each RGB frame, dynamic image and generated dynamic image. We sum them together and use average pooling in the temporal dimension to make the final prediction.

3.4 Experiments and results

In this section, we report a series of experiments to evaluate our action anticipation method. First, we present results for action recognition using the *static CNN* and the *dynamic CNN* in section 3.4.1. Then, we evaluate the impact of different loss functions for generating future dynamic images in section 3.4.2. After that in section 3.4.3, we compare our method with state-of-the-art techniques for action anticipation. Finally, in section 3.4.4 we present further analysis on the performance of our method through the iterative forecasting of new Dynamic Images and a visual comparison between the forecasted and computed Dynamic Images.

	JHMDB	UT-Interaction
Static CNN	55.0%	70.9%
Dynamic CNN	54.1%	71.8%

Table 3.1: Action recognition performance using dynamic and RGB images over JHMDB and UT-Interaction datasets. Action recognition performance is measured at frame level.

Datasets We test our method using three popular datasets for human action analysis JHMDB [Jhuang et al., 2013], UT-Interaction [Ryoo and Aggarwal, 2010] and UCF101-24 [Soomro et al., 2012], which have been used for action anticipation in recent prior works [Aliakbarian et al., 2017; Soomro et al., 2016b; Singh et al., 2017].

The **JHMDB** dataset is a subset of the HMDB51 dataset [Kuehne et al., 2011]. It is created by keeping action classes that involve a single person action. Videos have been collected from different sources such as movies and the world-wide-web. JHMDB dataset consists of 928 videos and 21 action classes. Each video contains one human action which usually starts at the beginning of the video. Following the recent literature for action anticipation [Aliakbarian et al., 2017], we report the average accuracy over the three splits and report results for so-called *earliest* setup. For the earliest recognition, action recognition performance is measured only after observing 20% of the video. To further understand our method, we also report recognition performance with respect to time (as a percentage).

The **UT-Interaction** dataset (UTI) contains 20 video sequences where the average length of a video is around 1 minute. These videos contain complete executions of 6 human interaction classes: shake-hands, point, hug, push, kick and punch. Each video contains at least one execution of interaction, and up to a maximum of 8 interactions. There are more than 15 different participants with different clothing. The videos are recorded with 30fps and with a resolution of 720 x 480 which we resize to 320 x 240. To evaluate all methods, we use recommended 10-fold leave-one-out cross-validation per set and report the mean performance overall sets.

The **UCF101-24** dataset is a subset of the challenging UCF101 dataset. This subset of 24 classes contains Spatio-temporal localisation annotation. It has been constructed for THUMOS-2013 challenge[2]. On average, there are 1.5 action instances per video; each instance cover approximately 70% of the duration of the video. We report the action-anticipation accuracy for set 1, as has been done previously by Singh et al. [2017].

3.4.1 Training of *Static* and *Dynamic* CNNs

In this section, we explain how we train our *static* and *dynamic* CNNs (see Figure 3.2). Similar to Bilen et al. [2016] and Bilen et al. [2017], we train a *Static CNN* for RGB frame-based video action recognition and a *Dynamic CNN* for dynamic image-based action recognition. In all our experiments, each dynamic image is constructed using

[2] http://crcv.ucf.edu/ICCV13-Action-Workshop/download.html

10 RGB frames (T=10). We use different data augmentation techniques to reduce the effect of over-fitting. Images are randomly flipped horizontally, rotated by a random amount in a range of -20 to 20 degrees, horizontally shifted in a range of -64 to 64 pixels, vertically shifted in a range of -48 to 48 pixels, sheared in a range of 10 degrees counter-clockwise, zoomed in a range of 0.8 to 1.2 and shifted channels in a range of 0.3. We make use of pre-trained Inception Resnet V2 [Szegedy et al., 2017] to fine-tune both *Static CNN* and the *Dynamic CNN* using a learning rate of 1e−4. We use a batch size of 32 and a weight decay of 4e−5. We use ADAM [Kingma and Ba, 2015] optimiser to train these networks using epsilon of 0.1 and beta 0.5. Action recognition performance using these CNNs for JHMDB and UTI datasets are reported in Table 3.1. Note that the action recognition performance in Table 3.1 is only at the frame level (not video level). We use these trained *Static* and *Dynamic* CNNs in the generation of future motion representation, dynamic images, and action anticipation tasks.

3.4.2 Impact of loss functions

In this section, we investigate the effectiveness of each loss function, explained in section 3.3.2, in the generation process of future dynamic images. We evaluate the quality of the generated dynamic images for the action recognition task. We feed the dynamic CNN, previously trained with *real dynamic images*, with the generated dynamic images, and we report the accuracy in the action recognition task.

We perform this experiment constructing a sequence of dynamic images using equation 2.7 for each test video in the dataset. Then for each dynamic image in the test set, we generate the future dynamic image using our convolutional autoencoder. Then we use our dynamic CNN (which has been pre-trained in the previous section) to evaluate the action recognition performance of generated dynamic images (**DIg**). Using this approach, we can evaluate the impact of several loss functions in the generation of dynamic images. Notice that the number of generated dynamic images is equal to real dynamic images in each dataset, which allow us to fairly compare the action recognition performance between the predicted and real dynamic images.

We use the first split of JHMDB and the first set of UTI to perform this experiment. We make use of the three proposed losses in section 3.3.2: dynamic-loss ($\mathcal{L}_{DL}$), class-based loss ($\mathcal{L}_{CL}$) and static-loss ($\mathcal{L}_{SL}$) to train our autoencoder. We train the convolutional autoencoder using ADAM solver with a batch size of 32, a learning rate of 1e−4. We train our model for 30 epochs using the same augmentation process used in section 3.4.1.

We use the generalisation performance of *real dynamic images* from Table 3.1 as a reference to estimate the quality of generated dynamic images since we measure the performance of generated dynamic images in the same way.

As can be seen in Table 3.2, a combination of $\mathcal{L}_{DL}$, $\mathcal{L}_{CL}$ and $\mathcal{L}_{SL}$ gives a good recognition performance in the case of JHMDB dataset obtaining 54.0% for the generated dynamic images, which as good as when we feed the real dynamic image to the *dynamic CNN* 54.1% (Table 3.1) — indicating that our generative model along

	JHMDB-21	**UT-Interaction**
$\mathcal{L}_{DL}$	42.8%	64.3%
$\mathcal{L}_{SL}$	49.5%	64.2%
$\mathcal{L}_{DL} + \mathcal{L}_{SL}$	53.4%	66.5%
$\mathcal{L}_{DL} + \mathcal{L}_{CL}$	52.5%	64.5%
$\mathcal{L}_{DL} + \mathcal{L}_{SL} + \mathcal{L}_{CL}$	54.0%	68.4%

Table 3.2: Results of using multitask learning to generate future dynamic images.

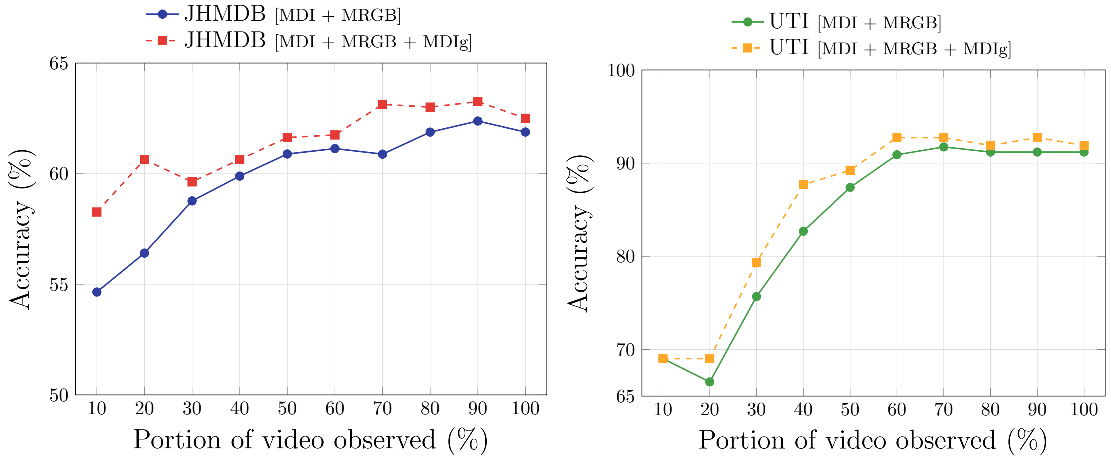


Figure 3.3: Action anticipation performance with respect to portion of the video observed on JHMDB *(left)* and UTI *(right)* datasets.

with the proposed loss functions is capable of generating representative and useful future dynamic images. A similar trend can be seen for UTI dataset. Notice that the $\mathcal{L}_{DL}$ and $\mathcal{L}_{SL}$ already produce good recognition performance on JHMDB and UTI datasets, which suggest that those losses can generate images that understand human motion. However, those generated images are not class-specific.

We conclude that convolutional autoencoder model trained with three losses is able to generate robust future dynamic images. These generated dynamic images are effective in action recognition.

3.4.3 Action anticipation

Our action anticipation network consist of a *static* CNN and a *dynamic* CNN (see Figure 3.2). Our action anticipation baseline uses observed multiple RGB frames and multiple dynamic images similar to Bilen et al. [2016]. In addition to that, our method generates a K future dynamic images and makes use of them with dynamic CNN. Action anticipation performance is evaluated at different time steps after observing fraction of the video (i.e., 10%, 20%, $\cdots$, 100% of the video). Results are shown in Figure 3.3, where we can see the effect of adding generated dynamic images (MDIg) to our pipeline. In the case of JHMDB, the most significant im-

Method	Earliest	Latest
S-SVN [Soomro et al., 2016a]	11.0%	13.4%
DP-SVM [Soomro et al., 2016a]	13.0%	14.6%
CuboidBayes [Ryoo, 2011]	25.0%	71.7%
CuboidSVM [Ryoo et al., 2010]	31.7%	85.0%
Context-Aware+Loss of [Jain et al., 2016a]	45.0%	65.0%
Context-Aware+Loss of [Ma et al., 2016a]	48.0%	60.0%
BP_SVM [Laviers et al., 2009]	65.0%	83.3%
I-BoW [Ryoo, 2011]	65.0%	81.7%
D-BoW [Ryoo, 2011]	70.0%	85.0%
E-LSTM [Aliakbarian et al., 2017]	84.0%	90.0%
Ours	89.2%	91.9%

Table 3.3: Comparison of action anticipation methods using **UTI** dataset. 50% of the video is observed at *Earliest*.

Method	Earliest	Latest
DP-SVM [Soomro et al., 2016a]	5%	46%
S-SVM [Soomro et al., 2016a]	5%	43%
Where/What [Soomro et al., 2016b]	12%	43%
Context-Aware+Loss of [Jain et al., 2016a]	28%	43%
Ranking Loss [Ma et al., 2016a]	29%	43%
Context-Aware+Loss of [Ma et al., 2016a]	33%	39%
E-LSTM [Aliakbarian et al., 2017]	55%	58%
ROAD [Singh et al., 2017]	57%	68%
Ours	**61%**	63%

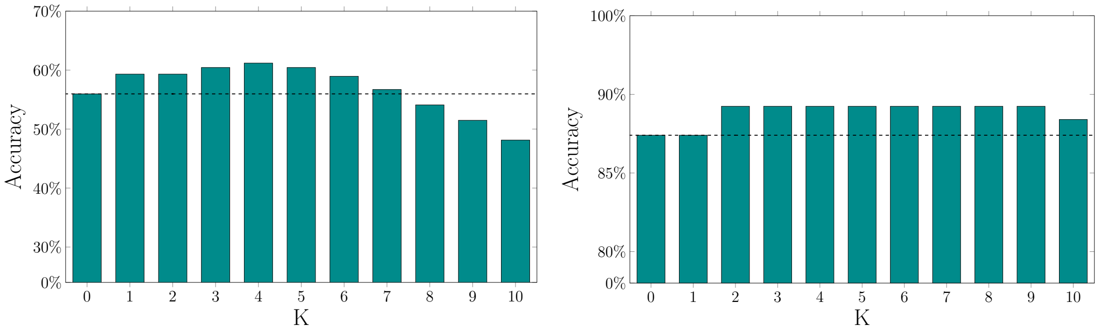
Table 3.4: Comparison of action anticipation methods on **JHMDB** dataset. 20% of video is observed at *Earliest*.

provement is obtained at 20%, which is an enhancement of **5.1%** with respect to the baseline. In the UTI dataset, the most significant improvement is obtained at 40% of the video observed with a performance enhancement of **5.0%** with respect to the baseline. Moreover, the less significant improvement is obtained when the video observation approaches the 100% with a 0.62% and 0.71% of improvement with respect to the baseline on JHMDB and UTI dataset, respectively.

Another standard practice is to report the action anticipation performance using *earliest* and *latest* prediction accuracies as done by Ryoo [2011]; Aliakbarian et al. [2017]. Although, there is no agreement of what is the proportion of frames used in the earliest configuration through different datasets. We make use of the proportion that has been employed by baselines (20% and 50% of the video for JHMDB and UTI, respectively). Therefore, following Aliakbarian et al. [2017] we report results in Tables 3.4 and 3.3 for JHMDB and UTI datasets, respectively. We outperform

	Earliest	Latest
ROAD (RTF) [Singh et al., 2017]	81.7%	83.9%
ROAD (AF) [Singh et al., 2017]	84.2%	85.5%
Ours	89.3%	90.2%

Table 3.5: Comparison of action anticipation methods on **UCF101-24** dataset. 10% of video is observed at Earliest.



(a) JHMDB

(b) UT-Interaction

Figure 3.4: Impact of generating more future dynamic images recursively on a) JHMDB and b)UT-Interaction datasets. K is the number of generated dynamic images based on observed RGB frames. K=0 means no dynamic image is generated.

other methods that rely on additional information, such as optical flow [Ma et al., 2016a; Soomro et al., 2016a,b] and Fisher vector features based on improved Dense Trajectories [Soomro et al., 2016a]. Our approach outperforms the state-of-the-art by **4.0%** on JHMDB and by **5.2%** on UTI datasets in the earliest configuration. Finally, we report results on UCF101-24 dataset for action anticipation. For this dataset, we use 10% of the video to predict the action class in the earliest configuration. As we can see in Table 3.5, we outperform previous method [Singh et al., 2017] by **5.1%** on the earliest configuration.

These experiments evidence the benefits of generating future motion information using our framework for action anticipation.

3.4.4 Further exploration

In Figure 3.4, we observe the influence of generating dynamic images recursively for the earliest configuration in JHMDB and UTI datasets. We generate K number of future dynamic images recursively. The first generation uses the very last true dynamic image in the video. Then, we use the previously generated dynamic image for the remaining $K-1$ generations. As it can be seen in Figure 3.4, as we generate more dynamic images into the future, the prediction performance degrades due to the error propagation. We report action recognition performance for each generated future dynamic image (i.e., for the generated future dynamic image at K). If we do not generate any dynamic image for the future, we obtain an action recognition

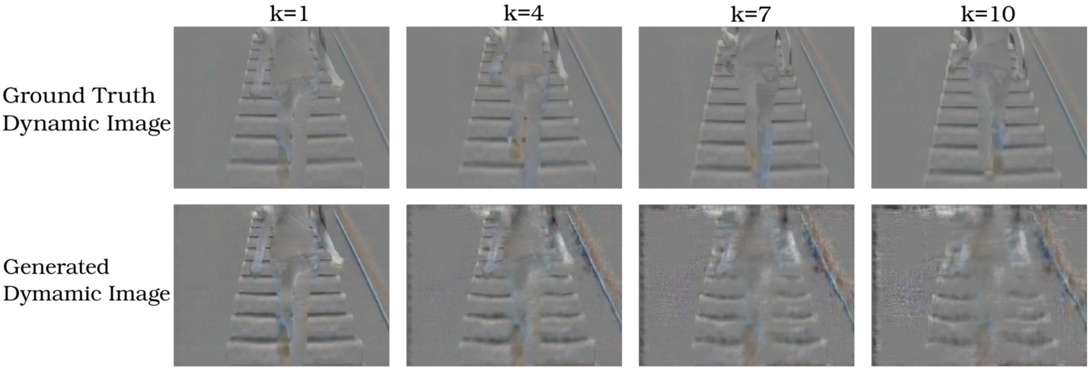


Figure 3.5: Visual comparison between generated dynamic image *(bottom)* and ground truth *(top)*. *K* refers to how many iterations we apply in the generation of dynamic image.

performance of 55.9%. If we include generated dynamic images, we obtain the best of 61.0% on JHMDB. A similar trend can be seen for UTI dataset, where without future dynamic images, we obtain 87.4%, and after generation, we obtain an action recognition performance of 89.2%. The influence of generating more future dynamic images is shown in Figure 3.4.

Finally, we visually inspect the recursively generated dynamic images for *K* equal to 1, 4, 7 and 10 in Figure 3.5. Although, we can use our model to generate quite accurate dynamic images, as we predict into the further, the generated dynamic images might contain some artifacts.

3.5 Summary

In this chapter, we have demonstrated how to hallucinate future video motion representation for action anticipation. We propose several loss functions to train our generative model in a multitask scheme. Our experiments demonstrate the effectiveness of our loss functions to produce better future video representation for the task of action anticipation. Moreover, experiments show that made use of the hallucinated future video motion representations improves the action anticipation results of our powerful backbone network. With our simple approach, we have outperformed the state-of-the-art in action anticipation in three important action anticipation benchmarks.

Chapter 4

Proposal-Free Temporal Moment Localisation Using Guided Attention

This chapter is based on

Rodriguez-Opazo, C.; Marrese-Taylor, E.; Sadat-Saleh, F.; Li, H; Gould, S. Proposal free Temporal Moment Localization of a Natural-Language Query in Video using Guided Attention. *Proceedings of the Winter Conference on Applications of Computer Vision*, 2020, pages 2453-2462.

Video analysis using natural language is drawing attention to the computer vision and natural language communities over the past few years, acknowledging the importance of these two modalities to understand the video content. In this chapter, we study the problem of temporal moment localisation in a long untrimmed video using natural language as the query. Given an untrimmed video and a query sentence, the goal is to determine the start and end of the relevant visual moment in the video that corresponds to the query sentence. In a vast majority of cases, these queries are descriptions of human actions. However, some of the benchmarks used for this task also have queries describing intrinsic temporal segments within a video such as 'credits'. While most previous works have tackled this by a propose-and-rank approach, we introduce a more efficient, end-to-end trainable, and proposal-free approach that is built upon three key components: First, we transfer language information to the visual domain using a dynamic filter. Second, we also propose a new loss function that guides the Dynamic Filter to attend the most relevant part of the video. Third, soft labels to cope with annotation uncertainties. Our method is evaluated on three standard benchmark datasets, Charades-STA, TACoS and ActivityNet-Captions. Experimental results show our approach outperforms state-of-the-art methods on these benchmarks datasets, confirming the effectiveness of our method.

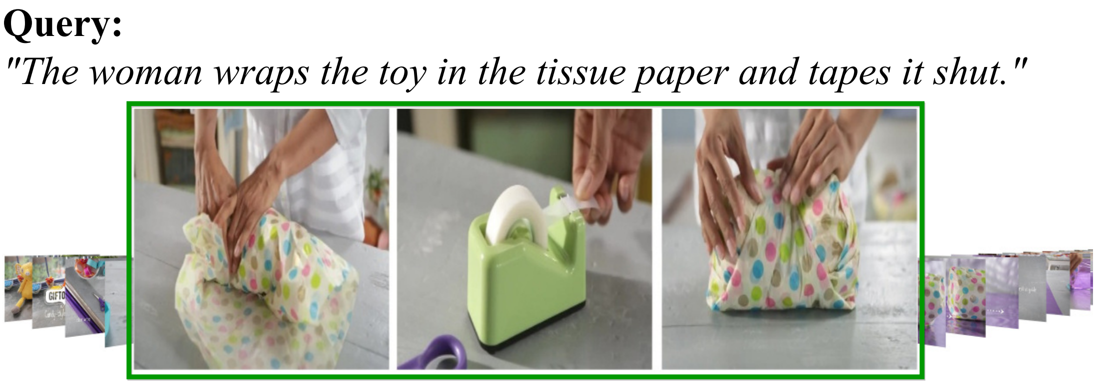

Query:
"The woman wraps the toy in the tissue paper and tapes it shut."

Figure 4.1: An illustration of temporal localisation of a natural language query in an untrimmed video. Given a query and a video the task is temporally localize the starting and ending of the sentence in the video.

4.1 Introduction

Vision-and-language understanding is an essential problem in computer vision, drawing increasing attention from the community over the past few years, motivated by its vast potential applications. This setting includes tasks such as video captioning and video question answering. While promising results have been achieved in these tasks, a fundamental issue remains to be tackled, namely, that these informative video segments need to be manually trimmed (pre-segmented) and often aligned with the relevant textual descriptions that accompany them.

Since searching for a specific visual event over a long video sequence is difficult and inefficient to do manually, even for a small number of videos, automated search engines are needed to deal with this requirement, especially when dealing with a massive amount of video data. It is clear that these search engines have to retrieve videos not only based on the video metadata, but they also must exploit their internal information in order to localise the required information/segment accurately.

In light of this, automatically recognising *when* an activity is happening in a video has become a crucial task in computer vision. Its applicability to other research areas such as video surveillance and robotics [Liu et al., 2019], among others, has also helped bring interest to this task. Earlier works in this area focused on *temporal action localisation* [Richard et al., 2018; Lin et al., 2017; Xu et al., 2017; Zhao et al., 2017; Escorcia et al., 2016b; Chao et al., 2018; Gao et al., 2017c], which attempted to localise "interesting" actions in a video from a pre-defined set of actions. However, this approach constrains the search engine to a relatively small and unrealistic set of queries from users.

To address this issue the task of "temporal action localisation with natural language" has been proposed recently [Gao et al., 2017a; Hendricks et al., 2017]. Given a query, the goal is to determine the start and end locations of the associated video segment in a long untrimmed video. In this context, we are specifically interested in the problem of natural-language-based temporal localisation, or temporal sentence localisation in the video. Formally, given an untrimmed video and a query in natural language, the task is to identify the start and endpoints of the video segment in response to it, therefore effectively locating the temporal segment (i.e., moment) that best corresponds to the given query, as depicted in Figure 4.1.

Current approaches to the localisation problem, either spatial or temporal, mainly focus on creating a good multi-modal embedding space and generating proposals based on the given query. In these *propose and rank* approaches, candidate regions are first generated by a separate method and then fed to a classifier to get the probabilities of containing target classes, effectively ranking them. Despite the relative success of these approaches, this setting is ultimately restrictive in scope since it uses predefined clips as candidates. These approaches are hard to extend for videos where the activities have a considerable variance in the length.

To this end, we propose an approach that does not rely on candidate generation or ranking, being able to directly predict the start and end times given a query in natural language. Our model is guided by a dynamic filter, which is responsible for matching the text and video feature embeddings, and a principal attention mechanism which encourages the model to focus on the features within of segment of interest. To the best of our knowledge, our approach is the first to do so.

Recent works on temporal action localisation with natural language Ghosh et al. [2019] has adopted an approach akin to machine reading comprehension [Chen et al., 2017], but in a multi-modal setting. Similar to ours, these models are trained in an end-to-end manner. Specifically, they maximise the likelihood of correctly predicting the start and end frames associated with a given query, analogous to predicting the text span corresponding to the correct answer in machine reading comprehension. We note, however, that annotating the start and end of a given activity in a video is highly subjective compare with machine reading comprehension, as evidenced by relatively low inter-annotator agreement [Sigurdsson et al., 2017; Alwassel et al., 2018]. In light of this, our model incorporates annotation subjectivity in a simple yet efficient manner, obtaining increased performance.

We conduct experiments on three challenging datasets, Charades-STA [Gao et al., 2017a], TACoS [Rohrbach et al., 2014] and Activity Net Captions [Krishna et al., 2017]. Our results demonstrate the effectiveness of our proposed method, obtaining state-of-the-art performance on those datasets. It also empirically demonstrate the effectiveness of our attention-based guidance mechanism, and of incorporating the subjective nature of the annotations into the model, ultimately validating our proposed approach through ablation analysis.

The rest of the chapter is organised as follows: Section 4.2 provides an overview of related work on temporal action localisation with natural language. Section 4.3 describes our method to find sentence in the video. Section 4.4 presents our experiments in two different benchmarks, quantitatively and qualitatively. Finally, in Section 4.5 we summarise our findings.

4.2 Related Work

4.2.1 Temporal Action Localization

The task of temporal action localisation aims to solve the problem of recognising and determining temporal boundaries of action instances in videos. Since activities (in the wild) consist of a diverse combination of actors, actions and objects over various periods of time, earlier work focused on the classification of video clips that contained a single activity, i.e., where the videos were manually trimmed.

More recently there has also been significant work in localising activities in longer, untrimmed videos. For example, Shou et al. [2016] trained C3D [Tran et al., 2015] with a localization loss and achieved state-of-the-art performance on THUMOS [Idrees et al., 2017]. On the other hand, Ma et al. [2016b] used a temporal LSTM to generate frame-wise prediction scores and then merged the detection intervals based on the predictions. Singh et al. [2016] extended the two-stream [Simonyan and Zisserman, 2014] framework with person detection and bi-directional LSTMs and achieved state-of-the-art performance on the MPII-Cooking dataset [Rohrbach et al., 2015].

Escorcia et al. [2016a] took a different approach and introduced an algorithm for generating temporal action proposals from long videos, which are used to retrieve temporal segments that are likely to contain actions. Lin et al. [2017] proposed an approach based on 1D temporal convolutional layers to skip the proposal generation step via directly detecting action instances in the untrimmed video, obtaining excellent results on the THUMOS [Idrees et al., 2017] and MEXaction2 [mex, 2015] datasets.

The major limitation of these action localisation methods is that they are restricted to a pre-defined list of actions. As it is non-trivial to design a label space which has enough coverage for such activities without losing useful details in users' queries, this approach makes it difficult to cover complex action queries.

4.2.2 Temporal language-driven moment localization

Language-driven temporal moment localisation is the task of determining the start and end time of the temporal video segment that best corresponds to a given natural language query. Essentially, this means using natural language queries to localise activities in untrimmed videos. While the language-based setting allows for having an open set of activities, it also corresponds to a more natural query specification, as it directly includes objects and their properties as well as the interaction between the involved entities.

The work of Hendricks et al. [2017] and Gao et al. [2017a] are generally regarded as a pioneering papers on this task. While Hendricks et al. [2017] proposed to learn a shared embedding for both video temporal context features and natural language queries, suitable for matching candidate video clips and language queries using a ranking loss and handcrafted heuristics, Gao et al. [2017a] proposed to generate candidate clips using temporal sliding windows which are later ranked based on alignment or regression learning objectives.

The research line defined by Gao et al. [2017a], where proposals are generated using temporal sliding windows was later extended by Ge et al. [2019], which leverage activity classifiers to help encode visual concepts, and add an *actionness score* to help capture the semantics from verb-object pairs in the queries. Recently, Liu et al. [2018] also resorted to sliding windows for generating proposals, but used a memory attention model when matching each proposal to the input query. Despite their simplicity and ability to provide coarse control over the frames that are evaluated, the main problem with these methods is that the matching mechanism between the candidate proposals and the query is computationally expensive.

To tackle this issue some approaches have focused on reducing the number of temporal proposals generated. These methods generally focus on producing query-guided or query-dependent video clip proposals, skipping unlikely clips from the matching step and thus speeding up the whole process. In this context, Chen et al. [2018] propose to capture frame-by-word interactions between video and language and then score a set of temporal candidates at multiple scales to localise the video segment that corresponds to the query. Similarly, Xu et al. [2019] propose a multilevel model to tightly integrate language and vision features and then use a fine-grained similarity measure for query-proposal matching.

A slightly different but related approach is proposed by Hendricks et al. [2018], where the video context is modelled as a latent variable to reason about the temporal relationships. The work of Zhang et al. [2019] further improved on this by utilising a graph-structured network to model temporal relationships among different moments, therefore addressing semantic and structural misalignment problems. On the other hand, Chen and Jiang [2019] focused on the proposal generation step, integrating the semantic information of the natural language query into the proposal generation process to get discriminative activity proposals. Although previous methods use techniques to directly generate candidate moment representations aligned with language semantics instead of fetching video clips independently, they still depend on ranking a fixed number of temporal candidates in each video, leading to inefficiencies, due to the long duration of videos with great variance in their lengths.

More recently, methods that go beyond the *scan and localise* approach, which can therefore directly output the temporal coordinates of the localised video segment have been proposed. For example, Yuan et al. [2019] used a co-attention based model for temporal sentence localization. In this context, some models also resort to reinforcement learning to dynamically observe a sequence of video frames conditioned on the given language query.

Concretely, Hahn et al. [2019] use this approach and learn how to skip around the video, therefore being able to more easily localise relevant clips in long videos. Instead of simply concatenating the video representation and query embedding, their approach uses a gated attention architecture to model textual and visual representations in order to align the text and video content.

Finally, Ghosh et al. [2019] proposed a simpler approach that does not rely on reinforcement learning and does not either involve retrieve and re-ranking multiple proposal segments. Their approach focuses on predicting the start and end frames

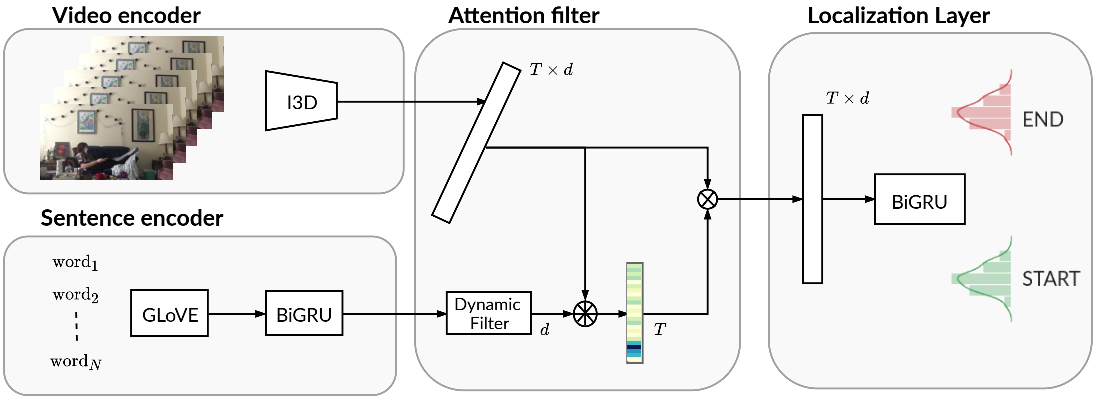


Figure 4.2: **Overview of our proposal-free method** with its four modules: sentence and video encoders to extract features from each modality, a dynamic filter to transfer language information to video, and a localization layer to estimate the starting and ending points. In the attention filter module, T is the total number of features extracted from the video and d is the dimensional space where the video and dynamic filter interact.

by leveraging cross-modal interactions between the text and video. In this context, our method proposes a simple yet effective proposal-free approach which makes it more practical to use.

4.3 Proposed Approach

Let $V \in \mathcal{V}$ be a video that can be characterized as a sequence of frames such that $V = [v_t \mid t = 1, \dots, \ell]$. Each video in $\mathcal{V}$ is annotated with a natural language query $Q \in \mathcal{Q}$ where Q is a sequence of words $Q = [w_j \mid j = 1, \dots, m]$, which describes what is happening in a certain period of time. Formally, this interval is defined by t^s and t^e, the starting and ending points of the annotations in time, respectively.

We propose a model that is trained end-to-end on a set of example tuples of annotated videos (V_k, Q_k, t^s_k, t^e_k). Although in the data a given video may be annotated with more than one single moment, and one natural language description may be associated to multiple moments, in this chapter we assume each derived case as an independent, separate training example. Given a new video and sentence tuple (V_r, Q_r), our model predicts the most likely temporal localisation of the contents of Q_r in terms of its start and end positions $t^{s\star}_r$ and $t^{e\star}_r$ in the video, therefore effectively solving the problem of temporal localisation of sentences in videos. In the following, for simplicity, we drop the index k associated to each training example.

Our model is designed in a modular way, offering more flexibility over previous work. There are four main components which we proceed to describe in the following sections. First, sections 4.3.1 and 4.3.2 give details about our video and natural language query encoders, respectively. These can be seen as the initial components in our model, responsible for effectively obtaining a semantically rich representation for the data coming from each input modality. The output representations returned by these modules are later combined using a dynamic filter layer, described in section 4.3.3, which allows us to transfer language information to the visual domain. Finally, section 4.3.4 describes our proposed localisation layer, which takes the filtered video features and uses them to predict the start and end frames of the desired location. Figure 4.2 shows an overview of our proposed approach.

4.3.1 Video Encoder

As discussed earlier, previous works on temporal sentence localisation in videos mostly rely on proposal generation, either using sliding windows or other heuristics [Gao et al., 2017a; Hendricks et al., 2017; Ge et al., 2019; Liu et al., 2018]. The process of producing many temporal segment candidates is computationally expensive, even though its efficiency can be improved if the proposals are processed in parallel. Moreover, proposal-based mechanisms neglect time dependencies across segments, treating them independently, thus ultimately failing to capture the temporal information in the input video effectively.

Inspired by recent works in one-shot object detection, we propose a video encoding layer that generates a visual representation summarising Spatio-temporal patterns directly from the raw input frames. Concretely, given an input video V, let $F_V(V)$ be our video encoding function mapping the l input video frames to a sequence of vectors $G = \{g_i \in \mathbb{R}^{d_v}\}$, $i = 1, \ldots, T$, with features that capture high-level visual semantics in the video. Note that the length of the input vector in frames l and the number of output features n may differ, which is why we denote them differently.

Because of the encoding of the video, the location of the annotated natural language description needs to be re-scaled to match the new feature-wise setting. We apply the mapping $\tau = (t \cdot n \cdot fps)/l$ to convert from frame/feature index to time. Concretely, t^s and t^e are converted into τ^s and τ^e corresponding to specific integer feature positions such that $\tau^s, \tau^e \in [1, \ldots, n]$.

Specifically, in this chapter, we model F_V using I3D proposed by Carreira and Zisserman [2017]. This method inflates the 2D filters of a well-known network, e.g., Inception [Szegedy et al., 2015; Ioffe and Szegedy, 2015] or ResNet [He et al., 2016] for image classification to obtain 3D filters, helping us better exploit the Spatio-temporal nature of the video. However, note that our video encoder later is generic, so other alternatives such as C3D propose by Tran et al. [2015] could be utilised instead.

4.3.2 Sentence Encoder

The language encoder aims at generating a semantically rich representation of the natural language query that is useful for localising relevant moments in the video. We model our encoder as a function $F_Q(Q)$ that maps each word w_j for $j = 1, \ldots, m$ to a semantic embedding vector $h_j \in \mathbb{R}^{d_s}$, where d_s defines the hidden dimension of the obtained sentence representation.

Although our sentence encoding module is generic, in this work, we rely on a bi-directional GRU [Chung et al., 2014] on top of pre-trained word embeddings.

Specifically, we make use of GloVe [Pennington et al., 2014], which are vectors pretrained in a large collection of text documents. In this setting, our query encoding function F is parametrised by both the GloVe embeddings and the GRU. Finally, to obtain a fixed-length sentence representation, we utilise a mean pooling layer over the hidden states obtained from the GRU, obtaining a final summarised query representation $\bar{h}$, as follows,

$$\bar{h} = \frac{1}{m} \sum_{j=1}^{m} h_j \tag{4.1}$$

4.3.3 Guided Attention

After encoding both the input sentence and video, we utilise an attention-based *dynamic filter* [Jia et al., 2016; Li et al., 2017; Gavrilyuk et al., 2018; Zhang et al., 2019]. The motivation behind this is to allow the model to generate filters to be applied over the video features that dynamically change given the sentence query, effectively reacting to specific parts of the video embedding and thus providing the model with clues about the location.

Concretely, we first reduce the dimensionality of the sentence embedding d^s and the video embedding d^v to the same space of size d using a fully-connected neural network f_s and f_v, respectively. Then, we apply a filter function θ as follows,

$$\theta(x) = tanh(W_\theta x + b_\theta) \in \mathbb{R}^T \tag{4.2}$$

As seen in Equation 4.2, our filter function $\theta(\cdot)$ is a single-layer fully-connected neural network. The sentence representation $\bar{h}$ is fed into our function and the obtained filter is later used to create a temporal attention over the projected video features G. Specifically, we apply a softmax over the dot product between each video feature g_i and the output of the filter $\theta(\bar{h})$, as follows,

$$A = softmax\left(\frac{f_v(G)^\intercal \theta(\bar{h})}{\sqrt{n}}\right) \in \mathbb{R}^T \tag{4.3}$$

$$\bar{G} = A \odot f_v(G) \in \mathbb{R}^{T \times d} \tag{4.4}$$

where $\odot$ denotes the Hadamard product, and the $1/\sqrt{n}$ constant is used to re-scale the product for better training stability [Vaswani et al., 2017]. As a result of these operations, each video feature is scaled by the attention filter based on the natural language query.

Given a category of semantically similar natural language queries, for example describing the same type of action, we would like our model to focus on the Spatio-temporal features that most likely describe and generalise these semantics across all examples where they are relevant, regardless of the additional context in the videos. We, therefore, argue that the most relevant features should fall inside the time boundary (τ_s to τ_e) defined by the starting and ending points of the target locations to be predicted. Although features from outside this segment could also contain useful information for the localisation task, we hypothesise that by exploiting these features the model should attain less generalisation power, as these features are not likely to capture patterns that appear in the majority of different videos containing a given type of action.

In light of this, we encourage our model to attend these relevant features and therefore improve its generalization capabilities. Concretely, we guide our attention mechanism to put focus on these features using a loss function on the output, as follows,

$$L_{att} = -\sum_{i=1}^{n}(1 - \delta_{\tau^s \leq i \leq \tau^e}) \log(1 - a_i) \tag{4.5}$$

where δ is the Kronecker delta and a_i is the ith column in the attention matrix A. In this way we penalise the network to attend features that are outside of the temporal moment refered by the query.

4.3.4 Localisation Layer

The localisation layer is in charge of predicting the starting and ending points of the moment in the video, using the previously constructed sequence of attended video features $\bar{g}_i$ for $i = 1, \ldots, n$.

Humans have difficulty agreeing on the starting and ending time of action inside a video, as evidenced by the low inter-annotation agreement in the relevant datasets for temporal localisation [Sigurdsson et al., 2017; Alwassel et al., 2018]. Considering that this is therefore a highly subjective task, we take a probabilistic approach and propose to use *soft-labels* [Salimans et al., 2016; Szegedy et al., 2016] to model the uncertainty associated with the labels.

The localisation layer starts by further contextualising the attended video features $\bar{g}_i$ utilising a 2-layer bidirectional GRU [Chung et al., 2014]. Then, we utilise two different fully connected layers to produce scores associated to the probabilities of each position being the start/end of the location. For each case, we take the softmax of these scores and thus obtain vectors $\hat{\boldsymbol{\tau}}^s, \hat{\boldsymbol{\tau}}^e \in \mathbb{R}^n$ containing a categorical probability distribution for the predicted start and end positions, respectively.

To model annotation uncertainty, we take τ^s and τ^e and create two target categorical distribution vectors $\boldsymbol{\tau}^s \sim \mathcal{N}(\tau^s, 1) \in \mathbb{R}^n$ and $\boldsymbol{\tau}^e \sim \mathcal{N}(\tau^e, 1) \in \mathbb{R}^n$ respectively, where $\mathcal{N}(\mu, \sigma)$ denotes a quantized Gaussian distribution centered at μ, with standard deviation σ. We train our model to minimize the Kullback-Leibler divergence between the predicted and ground truth probability distributions, as follows.

$$L_{KL} = D_{\mathrm{KL}}(\hat{\boldsymbol{\tau}}^s \parallel \boldsymbol{\tau}^s) + D_{\mathrm{KL}}(\hat{\boldsymbol{\tau}}^e \parallel \boldsymbol{\tau}^e) \tag{4.6}$$

where D_{KL} is the Kullback-Leibler divergence. The final loss for training our method is the sum of the two individual losses defined previously.

$$Loss = L_{KL} + L_{att} \tag{4.7}$$

During inference, we predict the starting and ending positions using the most likely locations given by the estimated distributions:

$$\hat{\tau}^s = \text{argmax}(\hat{\boldsymbol{\tau}}^s) \quad \hat{\tau}^e = \text{argmax}(\hat{\boldsymbol{\tau}}^e) \tag{4.8}$$

These values correspond to positions in the feature domain of the video, so we convert them back to time positions as explained previously.

4.4 Experiments

In this section, we first describe the datasets used in our experiments and give some details about our learning procedure. Then, we provide an ablation study on the effect of different components, and we compare our approach with other methods. Finally, we provide a qualitative visualisation of the predicted localisation and attention.

4.4.1 Datasets

To evaluate our proposed approach we work with three challenging datasets for temporal natural language-driven moment localisation, Charades-STA [Gao et al., 2017a], TACoS [Rohrbach et al., 2014] and ActivityNet Caption [Caba Heilbron et al., 2015; Krishna et al., 2017], which are widely utilised in previous works for evaluating models on our task.

Charades-STA: built upon the Charades dataset [Sigurdsson et al., 2016] which provides time-based annotations using a pre-defined set of activity classes, and general video descriptions. In Gao et al. [2017a], the sentences describing the video are semi-automatically decomposed into smaller chunks and aligned with the activity classes, which are later verified by human annotators. As a result of this process, the original class-based activity annotations are effectively associated to their natural language descriptions, totalling 13,898 pairs. We use the pre-defined train and test splits, containing 12,408 and 3,720 moment-query pairs respectively. Videos are 31 seconds long on average, with 2.4 moments on average, each being 8.2 seconds long on average.

MPII TACoS [Rohrbach et al., 2014] has been built on top of the MPII Compositive dataset [Rohrbach et al., 2012]. It consists of detailed temporally aligned text descriptions of cooking activities. The average length of the videos is five minutes. A significant challenge in TACoS dataset is that descriptions span over only a few seconds because of the atomic nature of queries such as 'takes out the knife' and 'chops the onion' (8.4% of them are less than 1.6s long). Such short queries allow a smaller margin of error. In total, there are 18,818 pairs of a sentence and video clips. We use the same split as in [Gao et al., 2017a], consisting of 50% for training, 25% for validation and 25% for testing.

ActivityNet Caption (ANet-Cap): a large dataset built on top of ActivityNet [Caba Heilbron et al., 2015], which contains data derived from YouTube and annotated for the tasks of activity recognition, segmentation and prediction. ANet-Cap further improves the annotations in ANet by incorporating descriptions for each temporal segment in the videos, totalling up to 100K temporal descriptions annotations over 20K videos. These have an average length of 2.5 minutes and are associated with over 200 activity classes, making the content much more diverse compared to Charades-STA. The temporally annotated moments are 36 seconds long on average, with videos containing 3.5 moments on average. Besides moments being more prolonged than in Charades-STA, we find that their associated natural language descriptions are also longer, besides using a more varied and richer vocabulary. We utilise the pre-defined train and validation splits in our experiments. Unlike Charades, Activity-Net contains a moment covering the entire video among other moments within the video.

Although other similar datasets, such as DiDeMo [Hendricks et al., 2017] also exist, we find it inadequate for evaluating our method. We note this dataset has been constructed for purposes that are substantially different from ours. They discretise videos into 5-second segments, and the task is to determine what is the start and end segment of the query. Thus, lacking start and end temporal annotations.

4.4.2 Implementation Details

Pre-processing for the natural language input in the case of Charades-STA is minimal, as we simply tokenise and keep all the words in the training data. In the case of ANEt-Cap, we pre-process the text with spacy[1] and replace all named entities as well as proper nouns with special markers. Finally, we truncate all sentences to a maximum length of 30 words and replace all tokens with a frequency lower than 5 in the corpus with a special *UNK* marker. The language encoder uses a hidden state of size 256, without fine-tuning the pre-trained GloVe embeddings.

When it comes to the video encoder, we first pre-process the videos by extracting features of size 1024 using I3D with average pooling, taking as input the raw frames of dimension 256 × 256, at 25 fps. For Charades-STA, we use the pre-trained model released by Carreira and Zisserman [2017] trained on Charades. For Anet-Cap we use the model pre-trained on the kinetics400 dataset [Kay et al., 2017] released by the same authors, which we also fine-tune on ANet-Cap afterwards.

All of our models are trained in an end-to-end fashion using Adam [Kingma and Ba, 2015] with a learning rate of 10^{-4} and weight decay 10^{-3}. To prevent over-fitting, we add dropout 0.5 between the two layers in the localisation module, which has a hidden size of 256. In addition to this, we also apply a simple data augmentation technique that consists on creating new examples by randomly cropping segments out from the initial part of the videos. We do this whenever the random cropping does not overlap with the locations of the annotations.

[1] https://spacy.io

4.4.3 Evaluation Metric

We evaluate our model by computing the temporal Intersection over Union (tIoU) at different thresholds, which we denote using the α parameter. In this setting, for a given value of α, whenever a given predicted time window has an intersection with the gold-standard that is above the α threshold, we consider the output of the model as correct. Following previous work, we also report the mean tIoU (mIoU) on the ANet-Cap dataset, helping make our comparisons more comprehensive.

4.4.4 Ablation Study

To evaluate the effectiveness of some introduced components, we perform several ablation studies on the Charades-STA dataset. Concretely, we focus on the soft-labeling technique and the usage of the attention loss L_{att}. For the latter we simply experiment omitting the term for the calculation of the gradients. For the former, we replace the L_{KL} loss with a likelihood-based loss similar to Ghosh et al. [2019], as follows:

$$L_{NLL} = -\log(\hat{\boldsymbol{\tau}}^s[\tau^s]) - \log(\hat{\boldsymbol{\tau}}^e[\tau^e]) \tag{4.9}$$

where $\hat{\boldsymbol{\tau}}^s$ and $\hat{\boldsymbol{\tau}}^e$ are the predicted probability distributions and τ^s and τ^e are the respective indices from the ground-truth annotations.

Method	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
NLL	60.91	43.66	27.07
KL	66.69	47.20	29.35
NLL + AL	66.64	47.53	29.89
KL + AL	**67.53**	**52.02**	**33.74**

Table 4.1: Ablation study on the impact of the guided attention and soft-labeling on Charades-STA.

We first compare our *soft-labeling* approach with the previously mentioned likelihood-based loss (NLL). As shown in Table 4.1, modelling the subjectivity of the labelling process using soft-labels and our distribution-matching loss (KL) leads to a significant improvement in the performance of our method, both in terms of retrieving and localising the full extent of the queries in the given videos.

We also evaluate the contribution of the attention loss L_{att} to our pipeline. According to the results in Table 4.1, adding the attention loss (AL) leads to consistent improvement in the performance of our method, both when modelling soft-labels and when not. This confirms our hypothesis that the most generalisable features are likely to be located within the boundaries of the query segment in the video. Finally, the synergy of our two proposed techniques can be seen in the last row of Table 4.1.

4.4.5 Comparison to the State-of-the-Art

We compare the performance of our proposed approach on both datasets against several prior work baselines.

Method	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
Random	-	8.51	3.03
CTRL [Gao et al., 2017a]	-	21.42	7.15
ABLR [Chen and Jiang, 2019]	-	24.36	9.01
SMRL[Wang et al., 2019]	-	24.36	11.17
SAP [Chen and Jiang, 2019]	-	27.42	13.36
MLVI [Xu et al., 2019]	54.70	35.60	15.80
TripNet [Hahn et al., 2019]	51.33	36.61	14.50
ExCL [Ghosh et al., 2019]	65.10	44.10	23.30
MAN [Zhang et al., 2019]	-	46.53	22.72
Ours	**67.53**	**52.02**	**33.74**

Table 4.2: Accuracy on Charades-STA for different tIoU α levels. Results for ABLR are as reported by Chen and Jiang [2019].

Method	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
MCN [Hendricks et al., 2017]	1.64	1.25	-
ABLR [Yuan et al., 2019]	18.90	9.30	-
CTRL [Gao et al., 2017a]	18.32	13.30	-
ACRN [Yuan et al., 2019]	19.52	14.62	-
TGN [Chen et al., 2018]	21.77	18.90	-
TripNet [Hahn et al., 2019]	23.95	19.17	9.52
MAC [Ge et al., 2019]	24.17	20.01	-
ExCL [Ghosh et al., 2019]	**44.20**	**28.00**	14.60
Ours	24.54	21.65	**16.46**

Table 4.3: Accuracy on TACoS for different intersection over union α levels.

Proposal-based methods: We compare our approach to a broad selection of models based on proposal generation, including MCN [Hendricks et al., 2017], TGN [Chen et al., 2018], MAN [Zhang et al., 2019], as well as some recent work such as SAP [Chen and Jiang, 2019], MLVI [Xu et al., 2019] and ACRN [Liu et al., 2018].

Reinforcement-learning-based methods: We compare our results to TripNet [Hahn et al., 2019] and SMRL [Wang et al., 2019], both of which utilise RL to learn how to jump through the video until the correct localisation is found.

Proposal-free methods: We consider two recent works, ABLR [Yuan et al., 2019] and ExCL [Ghosh et al., 2019], both aiming for proposal-free moment localization. Similar to ours, these techniques utilise the complete video representation to predict the start and end of a relevant segment. However, our approach is different since it models the uncertainty of the labelling process. Note also that while ABLR utilises a co-attention layer, ExCL does not rely on attention layers at all.

Comparing the performance of our method in the **Charades-STA** benchmark, our proposed approach outperforms all the baselines by a large margin, as can be seen in Table 4.2. The mean temporal intersection over union of our approach is 48.22, reflecting the capability of our method to correctly identify the correct temporal extent of the natural language query, as can also be seen in the performance at $\alpha = 0.7$

Method	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	mean tIoU
MCN [Hendricks et al., 2017]	42.80	21.37	9.58	-	15.83
CTRL [Gao et al., 2017a]	49.09	28.70	14.00	-	20.54
ACRN [Yuan et al., 2019]	50.37	31.29	16.17	-	24.16
MLVI [Xu et al., 2019]	-	45.30	27.70	13.60	-
TGN [Chen et al., 2018]	70.06	45.51	28.47	-	-
TripNet [Hahn et al., 2019]	-	48.42	32.19	13.93	-
ABLR [Yuan et al., 2019]	73.30	**55.67**	**36.79**	-	36.99
Ours	**75.25**	51.28	33.04	**19.26**	**37.78**

Table 4.4: Accuracy on ANet-Cap for different tIoU α levels

and $\alpha = 0.9$ where our method obtains 33.74 and 9.68 accuracy for those thresholds.

TACoS is a challenging benchmark not only because the length of the videos is much longer than Charades-STA, but also because it presents a bigger variability of segment duration for a query. The localisation layer in our system is a vanilla GRU, which could have difficulties predicting the precise spans of queries in very long videos, in the case of TaCoS videos can be as long as one hour. We believe that the variability of the moment durations in such long videos can harm the training process. Despite that, our method outperforms all previous methods at $\alpha = 0.7$, showing the robustness of our approach.

ANet-Cap is another challenging dataset similar to TACoS with significant variability of the duration of the segments. However, as shown in Table 4.4, our method yields good performance at different levels of tIoU. In particular, it outperforms all previous methods at $\alpha = 0.1$ and $\alpha = 0.7$, showing the effectiveness of our method to recall the correct temporal extent of the sentence query. Although our method cannot outperform the performance of ABLR at $\alpha = 0.3$ and $\alpha = 0.5$, it yields better mIoU than previous methods in this benchmark, as can be seen in Table 4.4. It is important to note that in this case, we do not compare with ExCL [Ghosh et al., 2019] since their reported results have more than 3,300 missing videos making those results not fair for comparison.

As suggested by the empirical evidence, our method consistently outperforms others on estimating the correct extension of the queries. This good estimation indicates that our proposed mechanism for incorporating the uncertainty of the labelling process is effective yet simple, playing a key role in helping the network to find the correct starting and ending points. In addition to this, the evidence also suggests that our novel attention mechanism, which guides the localisation layer to focus on the features that are within the corresponding segments in the video, also aids the network. By allowing the model to attend the features that better represent similar action across different videos, we obtain better generalisation.

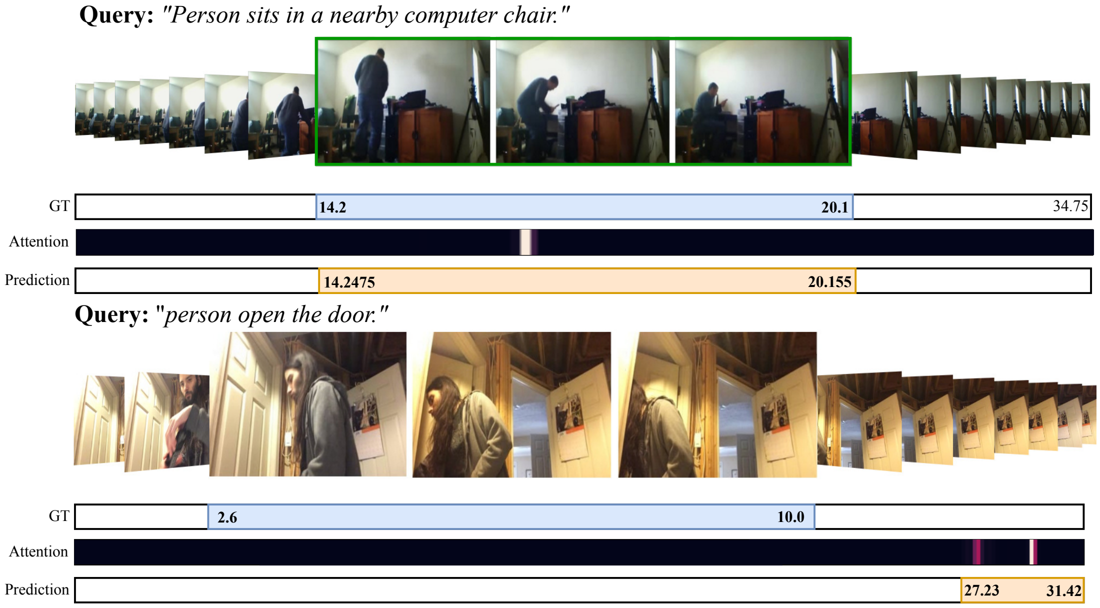


Figure 4.3: Examples of success and failure cases of our proposal-free method for Charades-STA

4.4.6 Qualitative Results

Two different samples one showing success and one a failure case of our method on Charades-STA dataset can be seen in Figure 4.3. Each sample presents the ground truth localisation, the attention weights and predicted localisation of a given query. For attention, brighter colours mean more weight. In the successful case, given the query *"Person sits in a nearby computer chair"*, our method could localise the moment at a tIoU of 98.28%, with maximum attention at 16.27 seconds peaking at 0.83. It is interesting to see that only one or two video features seem to be necessary for retrieving the starting and ending correctly.

On the second example in Figure 4.3 we show how our method fails to localise the query *"person open the door"*. It is possible to see that the appearance of the retrieved moment, when the person actually leaves the room, is very similar to the ground truth, Figure 4.4. We attribute this result to the features for opening the door and leaving the room being too close, especially on this example. Probably high-quality spatio-temporal features or deeper reasoning about the context would help the network to disambiguate this type of scenarios.

Other success cases of our algorithm on the **Charades-STA** dataset can be seen in Figures 4.5 and 4.6. It is interesting to see that as soon as our method can attend frames inside of the action, the localisation layer can predict a good start and end temporal location.

Failure cases of our method are presented in Figures 4.7 and 4.8. We can see that attention layer gets confused in the first example; it does not know what is the most

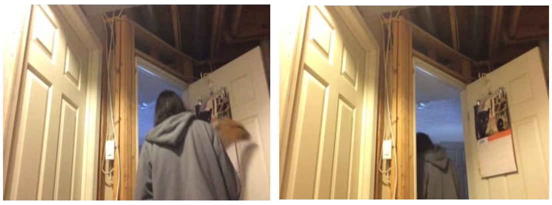


Figure 4.4: Similar appearance frames for failure case on Charades-STA

important feature for the query, making the localisation layer fail to predict good temporal localisation. Figure 4.8 shows that the attention layer gets confused with frames that has similar appearance, Figure 4.9.

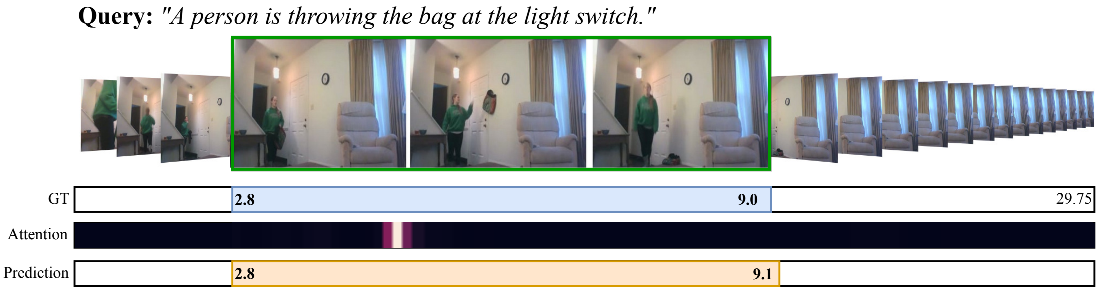


Figure 4.5: Success case of our proposal-free method on Charades-STA dataset

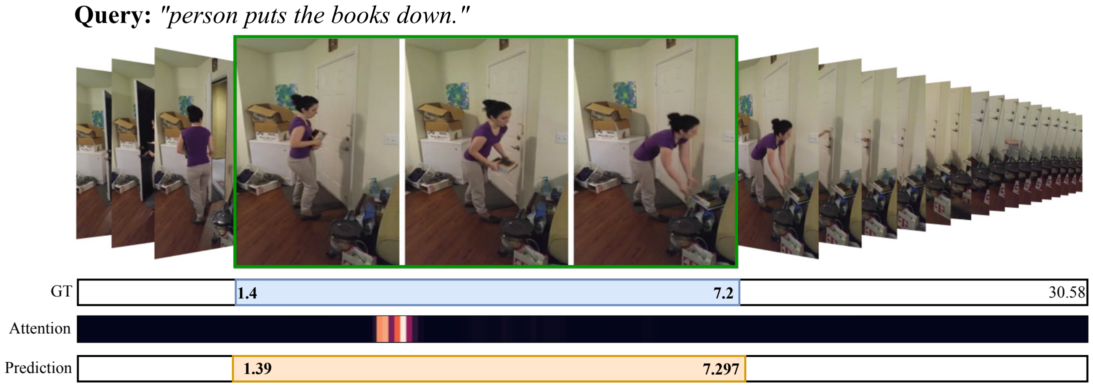


Figure 4.6: Success case of our proposal-free method on Charades-STA dataset

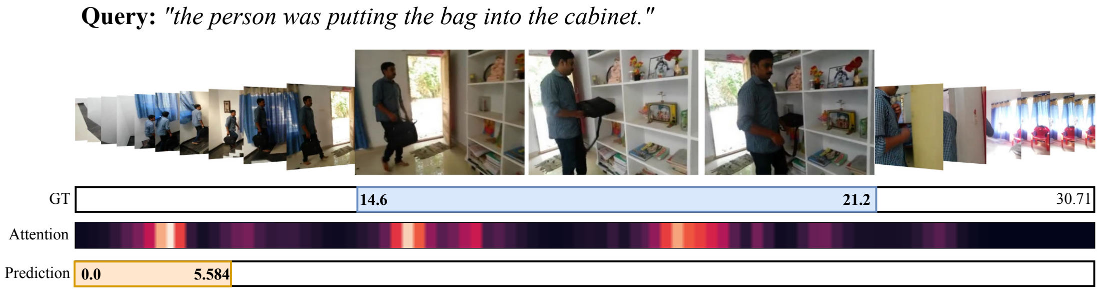


Figure 4.7: Failure case of our proposal-free method on Charades-STA

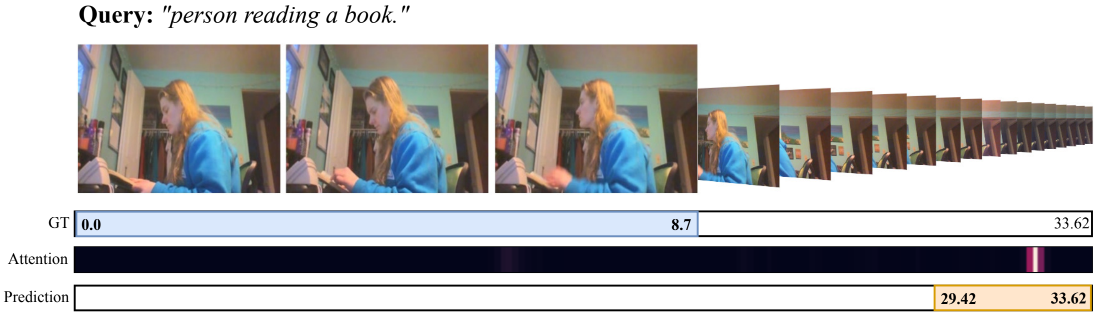


Figure 4.8: Failure case of our proposal-free method on Charades-STA

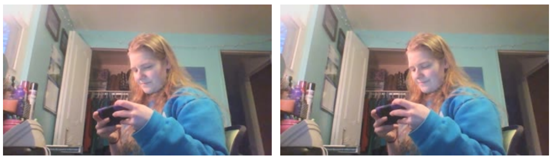


Figure 4.9: Confusing frames for one of the failure case of Charades-STA

Although videos in **ActivityNet Caption** are much longer than videos in Charades-STA, our method still can get good localisation performance if the attention layer does a good job, as can be seen in Figures 4.10 and 4.11. Notice that Figure 4.11 shows a long action that spans more than 2.5 minutes.

Failure cases of our method on ActivityNet Caption dataset are presented in Figures 4.12 and 4.13. Our method has similar difficulties in Charades-STA and ActivityNet Caption. Every time that the attention fails to focus in frames inside of the corresponding moment the localization layer cannot predict the correct temporal localization of the query. Figure 4.14 shows frames that are also related to query in Figure 4.13. These images suggest that our method can understand what a credits is and where is located but cannot distinguish *ending* or *starting*

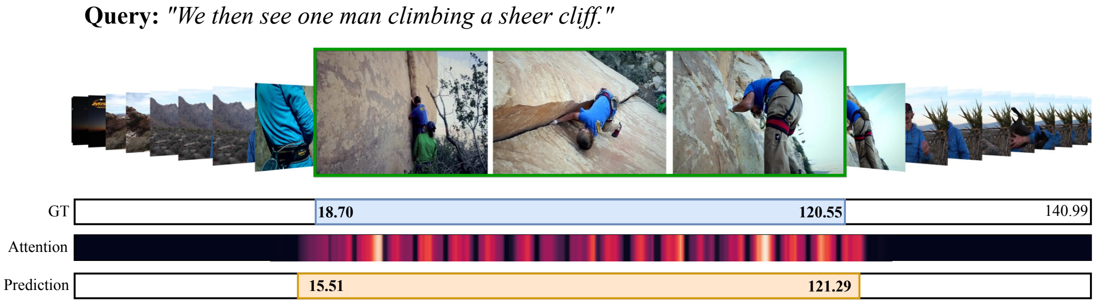


Figure 4.10: Success case of our proposal-free method in ActivityNet Caption dataset

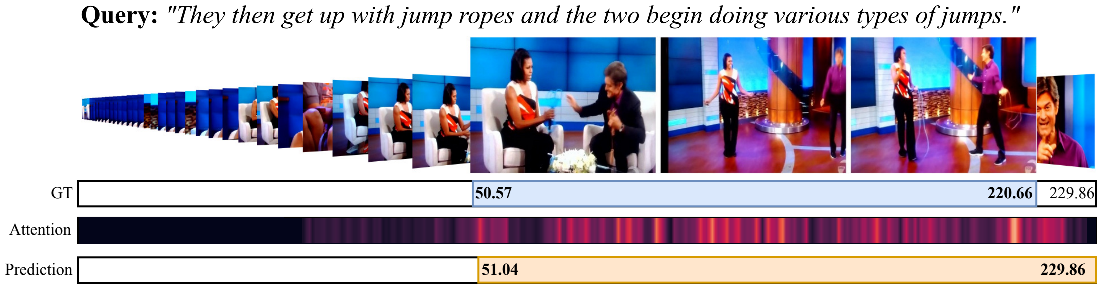

Query: "They then get up with jump ropes and the two begin doing various types of jumps."
GT
50.57
220.66
229.86
Attention
Prediction
51.04
229.86

Figure 4.11: Success case of our proposal-free method in ActivityNet Caption dataset

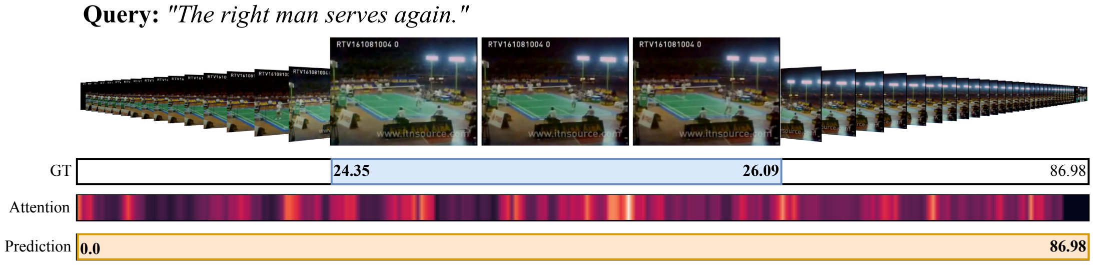


Figure 4.12: Failure case of our proposal-free method in ActivityNet Caption dataset

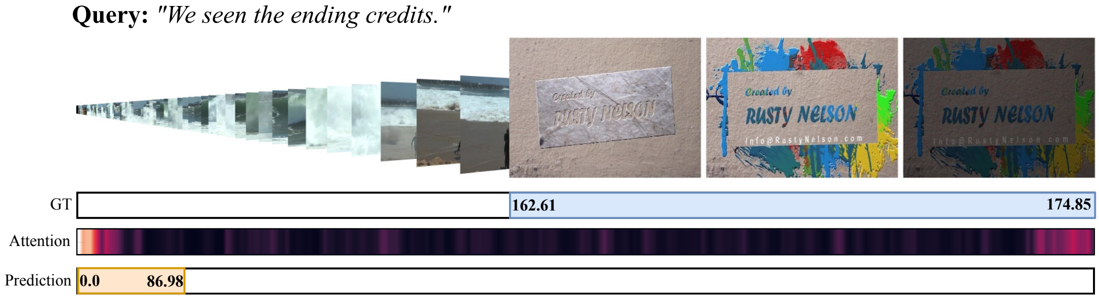


Figure 4.13: Failure case of our proposal-free method in ActivityNet Caption dataset.

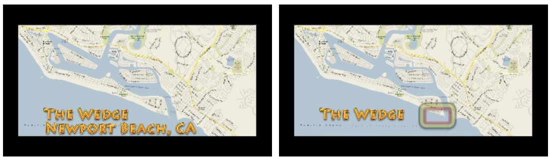


Figure 4.14: Confusing frames for one of the failure case of ActivityNet Caption

4.5 Summary

In this chapter, we have presented a novel end-to-end architecture that is designed to address the problem of temporal localisation of natural-language queries in videos, also known as *temporal moment localization*. Our approach uses a guided attention mechanism that focuses on more generalisable features to guide the localisation estimation. Moreover, we also consider the key problem of subjectivity in the annotation process by modelling the label uncertainty in a simple but efficient way, also obtaining substantial performance gains. As a result, our approach achieves state-of-the-art performance on both Charades-STA, TACoS and ANet-Cap datasets.

Chapter 5

Spatio-Temporal Graph for Language-based Query Localization in Video

This chapter is based on

Rodriguez-Opazo, C.; Marrese-Taylor, E.; Fernando, B.; Li, H; Gould, S. DORi: Discovering Object Relationships for Moment Localization of Natural Language Query in a Video. *Proceedings of the Winter Conference on Applications of Computer Vision*, 2021.

Localising actions in a long untrimmed video requires not just to understand the human motion involved in the execution of this, but also to understand the objects involved in the action and the sequence of actions performed previously. In the previous Chapter 4, we provide a solution for temporal moment localisation using a proposal-free approach and demonstrating the importance of attending generalisable temporal segments and how to deal with the uncertainty annotations. However, we have seen in Figures 4.4 and 4.9 how the method gets confused. Similar action such as "looking a mobile" or "reading a book" requires object information to understand the video adequately.

In this chapter, we continue studying the task of temporal moment localisation. However, we focus on adding spatial information, such as humans and objects involved in the video, to create a tailored activity representation for this task. Our key innovation is to learn a video feature embedding through a language-conditioned message-passing algorithm suitable for temporal moment localisation which captures the relationships between humans, objects and activities in the video. These relationships are obtained by a spatial subgraph that contextualised the scene representation using objects and human features detected by Faster-RCNN. Moreover, a temporal sub-graph captures the activities within the video through time. We evaluate our method on the same three standard benchmarks presented in the previous chapter, and we also introduce YouCookII as a new benchmark for this task. YouCookII is instructional videos that focused on the human and objects involved in the reproduction of a recipe. Experiments show our method outperforms state-of-the-art methods on these datasets, confirming the effectiveness of our approach.

5.1 Introduction

As the amount of video data continues to grow, searching for specific visual events in large video collections has become increasingly relevant for search engines. This search engine requirement has helped draw increased attention to the task of activity detection in recent years. This task is especially important, considering that manually annotating videos is laborious and error-prone, even for a small number of videos. In this sense, it is clear that search engines have to retrieve videos not only based on video metadata but that they must also consider the videos' content in order to localise a given query accurately.

It is for that reason that we retain our interest in the task of temporal sentence localisation, in which given an untrimmed video and a natural language query, the goal is to identify the start and end points of the video segment (i.e., moment) that best corresponds to the given query.

Many of the existing approaches to the localisation problem in vision-and-language, either spatial or temporal, have focused on creating a good multi-modal embedding space and generating proposals based on the given query. As it is mentioned in the previous chapter, in the propose and ranking methods, first a particular method generates candidate regions (proposals) which are then passed to a second stage that rank them and adjust the start and end locations of the query. Most recently, Ghosh et al. [2019] and our method presented in Chapter 4 do not rely on proposals.

Evidence shows that solving grounded language tasks often requires reasoning about relationships between objects in the context of the task [Hu et al., 2019]. For example, the work of Sigurdsson et al. [2017] showed that the performance in action recognition tasks improves by a large margin if we have a perfect object recognition oracle. Moreover, the majority of the queries that are used for this task are related to human actions. In this chapter, our primary motivation is to capture the relationship between humans and objects with the activity that they are performing. One can 'read a book' or 'look at the mobile.' A good way to identify and disambiguate what the person is doing is to make use of object clues.

In light of this, in this chapter, we propose a mechanism to obtain contextualised activity representations based on a language-conditioned message passing algorithm. As activities are usually the result of the composition of several actions or interactions between a subject and objects [Jiang et al., 2013], our algorithm incorporates both spatial and temporal dependencies. Therefore, modelling the relationship between subjects and objects in a scene and how these change over time, supporting the temporal moment localisation task.

We conduct experiments on four challenging datasets, Charades-STA [Gao et al., 2017a], ActivityNet [Caba Heilbron et al., 2015; Krishna et al., 2017], TACoS [Rohrbach et al., 2014] and YouCookII [Zhou et al., 2018b,a], demonstrating the effectiveness of our proposed method and obtaining state-of-the-art performance. Our results highlight the importance of our message-passing algorithm in modelling the relationship between human and object and their interaction to understand the activity, ultimately validating our proposed approach. Our approach is the first to incorporate a language-conditioned message-passing algorithm to obtain contextualised activity representations using the objects and subjects for this task to the best of our knowledge.

5.2 Related Work

In the same manner that chapter 4, this work is related to the temporal action localisation task, which aims to recognise and determine the temporal boundaries of action instances in videos. There is extensive previous work on this task, ranging from models that train existing video feature extractors with a localisation loss [Shou et al., 2016], to systems that generally rely on temporal action proposal, as well as more sophisticated models that perform contextual modelling, capturing objects and their interactions [Gu et al., 2018; Girdhar et al., 2019].

After the introduction of the AVA dataset [Gu et al., 2018], which contains clips labelled with people and their actions, various proposals have attempted to perform contextual modelling, i.e. capturing objects and object interactions for action localisation. This contextual modelling is the case of Girdhar et al. [2019], who recently proposed a Transformer-based model which uses its attention mechanism, learns to emphasise hands and faces, which are often crucial to discriminate an action.

Since action localisation is restricted to a pre-defined list of options, Gao et al. [2017a] and Hendricks et al. [2017] introduced a variation of the task called language-driven temporal moment localisation. Essentially, this means to use natural language queries to localise activities in untrimmed videos. While the language-based setting allows for having an open set of activities, it also corresponds to a more natural query specification, as it directly includes objects and their properties as well as relations between the involved entities.

Early approaches for this task, including Liu et al. [2018] and Ge et al. [2019], were mainly based on generating proposals or candidate clips which could later be ranked. More recently, Chen et al. [2018], Chen and Jiang [2019], and Xu et al. [2019], have worked on reducing the number of proposals by producing query-guided or query-dependent approaches.

Despite their ability to provide coarse control over the video snippets, proposal-based methods suffer from the computationally expensive candidate proposal matching, which has led to the development of methods that can directly output the temporal coordinates of the segment. In this context, Yuan et al. [2019] first proposed to use a co-attention-based model, and soon after Ghosh et al. [2019] focused directly on predicting the start and end frames using regressions. More recently, our work in Chapter 4 used dynamic filters and modeled label uncertainty to further improve performance [Rodriguez-Opazo et al., 2020], while Mun et al. [2020] and Zeng et al. [2020] proposed more sophisticated modality matching strategies. Compared to these works, although our approach is also proposal-free, we differ in the sense that we aim at incorporating specific spatial information that is useful for the localisation problem.

This chapter is also related to context modelling in action recognition. In this context, structural-RNN [Jain et al., 2016b] models a spatio-temporal graph using an RNN mixture that is differentiable, with applications on human motion modelling and human activity detection. While we build on top of a concept similar to this, we inject the language component into the spatio-temporal graph and focus on the task of temporal moment localisation of a natural language query. Our method adds the language into the pipeline using an attention mechanism that captures the objects' and subjects' interactions at the language level.

Context modeling has also been recently utilized in other computer vision tasks, such as referring expression comprehension [Yu et al., 2018] and VQA [Hu et al., 2019]. In the latter, the authors proposed a Language-Conditioned Graph Network (LCGN) where each node represents an object and is described by a context-aware representation from related objects through iterative message-passing conditioned on the textual input. Our work is fundamentally different from this as our task requires us to model the temporal component in our graph. Moreover, LCGN emphasises the role of edge representations in the graph whilst our approach is node-centric as connections between two given node types share the same edges.

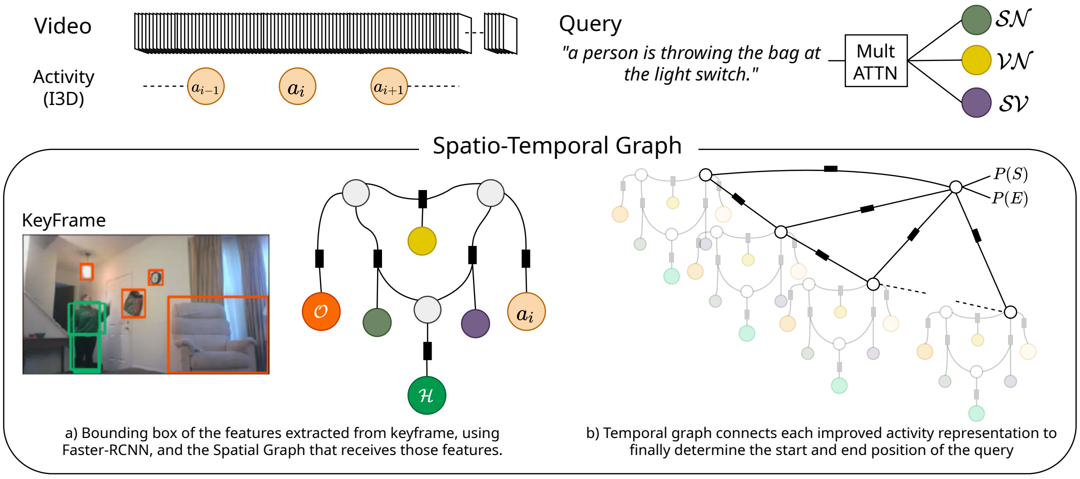
Furthermore, Zeng et al. [2019] used graph convolutions to obtain contextualised representations for action localisation, while Zhang et al. [2019] utilised a graph-structured network to model temporal relationships among different moments and thus obtain contextualised moment representations. Their approach is different from ours as they rely on proposals to perform the task. More recent approaches, such as SLTFNet [Jiang et al., 2019], rely on attention instead of message-passing to deal with the spatio-temporal nature of the moment localisation task.

Finally, Zhang et al. [2020] have recently proposed a novel task that requires not only to perform temporal language-driven moment localisation but also to locate the objects mentioned in the query spatially. Their approach is similar to ours in the sense that it also utilises a spatio-temporal graph. However, the textual clues are incorporated after the graph construction rather than being an explicit part of it.

5.3 Graph Based Temporal Moment Localization

In temporal moment localisation, the objective is to find the temporal location of a natural language query Q in an untrimmed video V. The video consists of a sequence of frames $V = [v_t \mid t = 1, \ldots, \ell]$ and the query is a sequence of words $Q = [w_j \mid j = 1, \ldots, m]$ that describes a short moment in the video. We denote the starting and ending times of the moment described by query Q as t^s and t^e, respectively.

We propose a model that explicitly captures the relationship between objects and humans, as well as the activities performed in a video, using a spatio-temporal graph. Concretely, we utilise a language-conditioned message-passing algorithm, which allows us to obtain contextualised activity representations for better moment localisation. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}_S \cup \mathcal{E}_T)$ represent our spatio-temporal graph, where $\mathcal{V}$, $\mathcal{E}_S$ and $\mathcal{E}_T$



a) Bounding box of the features extracted from keyframe, using Faster-RCNN, and the Spatial Graph that receives those features.

b) Temporal graph connects each improved activity representation to finally determine the start and end position of the query

Figure 5.1: **Overview of our spatio-temporal graph** for temporal moment localisation: For each activity feature a_i, we create a Spatial graph to find the relationship between object and human nodes conditioned in the query, and thus improve the activity representation to be used by the Temporal graph.

are the set of nodes, spatial edges and temporal edges, respectively, as can be seen in Figure 5.1.

We factorize our spatio-temporal graph into spatial and temporal sub-graphs, denoted by $\mathcal{G}_S = (\mathcal{V}, \mathcal{E}_S)$ and $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T)$, respectively.

The spatial graph is designed to improve the *activity representations* by exploiting the relationships between objects and humans in a given scene conditioned on an attended language representation for each of this relationships. As we know [Sigurdsson et al., 2017], actions and moments are characterised by complex interactions between humans as well as human-object interactions. Our spatial sub-graph is designed to exploit these spatial relationships specifically. It is iteratively applied through the video (Figure 5.1.a.)

On the other hand, the temporal sub-graph is designed to model the relationships between the improved activity representations at different times to more efficiently localise the start and end points of the query in the video (Figure 5.1.b.)

5.3.1 Spatial graph

Consider the query presented in Figure 5.1, *"a person is throwing the bag at the light switch"*. It describes what action (*verb*) is performed by a *subject* and what *objects* are involved in that action. Our spatial graph is designed to capture the relationships between these visual entities conditioned on the linguistic entities. As such, we decompose our graph into six semantically meaningful nodes, three for representing visual information and three for representing linguistic information.

5.3.1.1 Linguistic Nodes

We create language nodes to capture essential information in the query related to the visual input: We expect but not enforce that these nodes captures the *subject-verb* relationship node $\mathcal{SV}$ (person-throwing), the *subject-object* relation node $\mathcal{SN}$ (person-bag/light switch) and the *verb-object* relation node $\mathcal{VN}$ (throwing-bag/light switch). Therefore, there is no need to extract such information from the sentence using methods like dependency parsing. We let the attention mechanisms decide the most relevant words for each node.

To obtain representations for each one of the linguistic nodes, we start by encoding each of the words w_j for $j = 1, \ldots, m$ in the query Q using a function $F_w : w \mapsto h$, which maps each word to a semantic embedding vector $h_j \in \mathbb{R}^{d_w}$, where d_w defines the hidden dimension of the word embedding. Specifically, we use GLoVe embeddings [Pennington et al., 2014] to obtain the vector representations for each word.

We then initialise three-headed multi-head attention module [Vaswani et al., 2017] using an aggregated, fixed-length query vector q. Concretely, we construct this vector using a bi-directional GRU [Chung et al., 2014] over the word embeddings and mean pooling, which allows us to more accurately capture the global meaning of the input query by first contextualising each word representation. We project each of the word embeddings using a linear mapping to obtain the key k components of multi-head attention. In the case of the values v we use the contextualised word representations from the GRU. Each head attends these contextualised vectors and returns a re-weighted combination of them, using $\text{softmax}(qk^\top)v$ and aimed at understanding a specific relation between the visual nodes at the linguistic level.

5.3.1.2 Visual Nodes

As mentioned above, our spatial graph contains three semantically meaningful nodes that represent visual information, specifically an activity node $\mathcal{A}$, a human node $\mathcal{H}$ and an object node $\mathcal{O}$. This setting allows us to share factors for semantically similar observations taken from the video [Jain et al., 2016b], which provides several advantages. First, the model can deal with more observations of objects and humans without increasing the number of parameters that need to be learnt. Second, we alleviate the problem of having jittered object detections in videos, especially due to objects appearing and disappearing across frames.

To capture the relationships between activity, human and object observations, we densely connect these nodes within a single video frame. Such relationships are commonly parameterised by factor graphs that convey how a function over the graph factorises into simpler functions [Kschischang et al., 2001]. Similarly, we learn a non-linear mapping function for each of the semantically alike observations that are associated with the same semantic node. In this sense, each semantic node, human $\mathcal{H}$, object $\mathcal{O}$ and activity $\mathcal{A}$, is considered to be a latent representation of the corresponding observation. Let us take as an example the case of the object node $\mathcal{O}$, where we observe a *table* in the video, represented by a feature vector x, obtained directly from the object detector. In this case, we use a function,

$$\Psi_{\mathcal{O}} \doteq \tanh(W_{\mathcal{O}} x + b_{\mathcal{O}}) \tag{5.1}$$

Similar mapping functions (with different parameters), namely $\Psi_{\mathcal{H}}$ and $\Psi_{\mathcal{A}}$ are defined for the other semantic nodes.

Activity node: We use a video encoder that generates a video representation summarizing spatio-temporal patterns directly from the raw input frames. Concretely, let F_V be our video encoding function that maps a video into a sequence of vectors $[a_i \in \mathbb{R}^{d_v} \mid i = 1, \ldots, T]$. These features capture high-level visual semantics in the video. Note that length of the video, $\ell = |V|$, and the number of output features, $t = |F_V(V)|$, are different due to temporal striding. As in the previous Chapter 4, we model F_V using I3D [Carreira and Zisserman, 2017]. This method inflates the 2D filters of a well-known convolutional neural network, e.g., Inception [Szegedy et al., 2015; Ioffe and Szegedy, 2015] or ResNet [He et al., 2016] for image classification to obtain 3D filters.

Human and object nodes: Activity representations are obtained using small clips of frames. This means that there may be a set of many frames from where to extract spatial information that is semantically relevant for each node. Utilising every frame is computationally expensive, and given the piece-wise smooth nature of the video, this could also prove to be redundant. As such, in this work, we propose to utilise key-frames associated with each activity representation to extract observations for human and object nodes. Since many frames in a video are blurry due to various reasons, e.g., the natural movement of the objects and the camera motion, we select the sharpest key-frame in the subset of frames. Here we use the Laplace variance algorithm [Pech-Pacheco et al., 2000], which is a well-known approach for measuring the sharpness of an image.

While our method is agnostic to the choice of object detector, in this work we use Faster RCNN [Ren et al., 2015; Anderson et al., 2018] for the detection and spatial representation of the objects in all key-frames. Our Faster RCNN detector is trained on the Visual Genome [Krishna et al., 2016] dataset, which consists of 1,600 object categories. These categories are manually assigned to either the human and object nodes depending on the type of object. The human node receives the set of features $H = \{h_1, ..., h_K\}$ corresponding to the categories associated to human body parts, clothes and subjects, while the object node receives the set of features that are not associated to human labels with that $O = \{o_1, ..., o_J\}$. This label-based categorisation is based on a manual analysis of the label names supported by the Faster RCNN detector. In this way, when taking the predicted labels for each object, we can use our categorisation to re-label them as human or object and thus assign each instance to their corresponding visual node.

5.3.1.3 Language-conditioned message-passing

We argue that the setting of the scene contains important clues to improve the representation of a given activity. Examples of these clues are human clothes, objects that are present in the scene as well as their appearance. To the best of our knowledge, previous work on moment localisation has not utilised this information. Therefore, we propose to obtain an activity representation suitable for the moment localisation task, by capturing object, human and activity relationships. Concretely, we use a mean-field like approximation of the message-passing algorithm to capture such relationships. The messages sent between nodes are conditioned on the natural language query. We propose to use this approximation instead of the standard message-passing algorithm due to high demand on memory and compute, especially to process all the key-frames in a given video. The messages are iteratively sent a total of N times, which is a hyperparameter of our model. In the equations below, index $n = 1, ..., N$ denotes the iteration step for each of the nodes. Notice that in the rest of this subsection, we drop the temporal index i in the activity feature a since the message-passing is done for each of the activity features independently.

First, we capture the relationship between the visual observations of the nodes human $\mathcal{H}$, object $\mathcal{O}$ and activity $\mathcal{A}$ with the corresponding language nodes $\mathcal{SN}, \mathcal{SV}$ and $\mathcal{VN}$ that connect the semantic meaning of the visual nodes, using a linear mapping function f specific for each node. The input for this function is the concatenation of the language with the visual observation. For instance, in the case of the object observations, the mapping functions f have the following shape.

$$f_{\mathcal{SN},\mathcal{O}}(\mathcal{SN}, o^{j,n}) = W_{sno}[\mathcal{SN}; o^{j,n}] + b_{sno} = \Phi^{j,n}_{\mathcal{SN},\mathcal{O}} \tag{5.2}$$

$$f_{\mathcal{VN},\mathcal{O}}(\mathcal{VN}, o^{j,n}) = W_{vno}[\mathcal{VN}; o^{j,n}] + b_{vno} = \Phi^{j,n}_{\mathcal{VN},\mathcal{O}} \tag{5.3}$$

where W_{sno} and W_{vno} are learnable linear projections, ";" represents the concatenation between the features and j is the j-th object observation in the object node $\mathcal{O}$. Similarly, we have specific mapping functions

$$f_{\mathcal{SV},\mathcal{A}}(\mathcal{SV}, a^{n}) = \Phi^{n}_{\mathcal{SV},\mathcal{A}} \quad f_{\mathcal{VN},\mathcal{A}}(\mathcal{VN}, a^{n}) = \Phi^{n}_{\mathcal{VN},\mathcal{A}}, \tag{5.4}$$

$$f_{\mathcal{SN},\mathcal{H}}(\mathcal{SN}, h^{k,n}) = \Phi^{k,n}_{\mathcal{SN},\mathcal{H}} \quad \text{and} \quad f_{\mathcal{SV},\mathcal{H}}(\mathcal{SV}, h^{k,n}) = \Phi^{k,n}_{\mathcal{SV},\mathcal{H}} \tag{5.5}$$

for the activity and human observations, where k is the k-th human observation.

For clarity, we explain the message-passing algorithm again using the object node as an example. The object node $\mathcal{O}$ receives messages from the human $\mathcal{H}$ and the activity $\mathcal{A}$ nodes. The message from the human node is constructed using a linear mapping function that receives as an input the concatenation of the object-query relationship $\Phi^{j,n}_{\mathcal{SN},\mathcal{O}}$ and the aggregation of all the human-query relationships $\sum_k \Phi^{k,n}_{\mathcal{SN},\mathcal{H}}$. A similar process is done for the message received from the activity observations, as can be seen in Equation 5.7, below.

$$\Psi^{j,n}_{\mathcal{H},\mathcal{SN},\mathcal{O}} = f_{\mathcal{H},\mathcal{SN},\mathcal{O}}(\Phi^{j,n}_{\mathcal{SN},\mathcal{O}}, \textstyle\sum_{k=1}^{K} \Phi^{k,n}_{\mathcal{SN},\mathcal{H}}) \tag{5.6}$$

$$\Psi^{j,n}_{\mathcal{A},\mathcal{VN},\mathcal{O}} = f_{\mathcal{A},\mathcal{VN},\mathcal{O}}(\Phi^{j,n}_{\mathcal{VN},\mathcal{O}}, \Phi^{n}_{\mathcal{VN},\mathcal{A}}) \tag{5.7}$$

$$o^{j,n+1} = \sigma(m_o(\Psi^{j,n}_{\mathcal{H},\mathcal{SN},\mathcal{O}} \odot \Psi^{j,n}_{\mathcal{A},\mathcal{VN},\mathcal{O}}) \odot o^{j,0}) \tag{5.8}$$

Finally, the new representation of the object observation is computed using Equation 5.8, where $o^{j,0}$ is the initial object representation, σ is an activation function, $\odot$ is the Hadamard product and m_o is a linear function with a bias that constructs the message for the object o^j. A similar process is applied for each observation, as can be seen in Equations 5.9 to 5.12 below, where we create the message for each edge and Equations 5.13 to 5.14 and show how these messages are later used to contextualize the features. Note that the parameters learnt for each specific case are shared. For instance, parameters for $f_{\mathcal{A},\mathcal{SV},\mathcal{H}}$ and $f_{\mathcal{H},\mathcal{SV},\mathcal{A}}$ are the same.

$$\Psi^{n}_{\mathcal{H},\mathcal{SV},\mathcal{A}} = f_{\mathcal{H},\mathcal{SV},\mathcal{A}}(\Phi^{n}_{\mathcal{SV},\mathcal{A}}, \textstyle\sum_{k=1}^{K} \Phi^{k,n}_{\mathcal{SV},\mathcal{H}}) \tag{5.9}$$

$$\Psi^{n}_{\mathcal{O},\mathcal{VN},\mathcal{A}} = f_{\mathcal{O},\mathcal{VN},\mathcal{A}}(\Phi^{n}_{\mathcal{VN},\mathcal{A}}, \textstyle\sum_{j=1}^{J} \Phi^{j,n}_{\mathcal{VN},\mathcal{O}}) \tag{5.10}$$

$$\Psi^{k,n}_{\mathcal{O},\mathcal{SN},\mathcal{H}} = f_{\mathcal{O},\mathcal{SN},\mathcal{H}}(\Phi^{k,n}_{\mathcal{SN},\mathcal{H}}, \textstyle\sum_{j=1}^{J} \Phi^{j,n}_{\mathcal{SN},\mathcal{O}}) \tag{5.11}$$

$$\Psi^{k,n}_{\mathcal{A},\mathcal{SV},\mathcal{H}} = f_{\mathcal{A},\mathcal{SV},\mathcal{H}}(\Phi^{k,n}_{\mathcal{SV},\mathcal{H}}, \Phi^{n}_{\mathcal{SV},\mathcal{A}}) \tag{5.12}$$

$$a^{n+1} = \sigma(m_a(\Psi^{n}_{\mathcal{H},\mathcal{SV},\mathcal{A}} \odot \Psi^{n}_{\mathcal{O},\mathcal{VN},\mathcal{A}}) \odot a^{0}) \tag{5.13}$$

$$h^{k,n+1} = \sigma(m_h(\Psi^{k,n}_{\mathcal{O},\mathcal{SN},\mathcal{H}} \odot \Psi^{k,n}_{\mathcal{A},\mathcal{SV},\mathcal{H}}) \odot h^{k,0}) \tag{5.14}$$

5.3.2 Temporal graph

The temporal graph is responsible for predicting the starting and ending points of the moment in the video. It uses the previously computed activity representations $a^{i,N}$ for $i = 1, \dots, t$ where N is the final iteration in the message passing. The temporal graph is implemented using a 2-layer bi-directional GRU [Cho et al., 2014], which receives as input the improved activity representation, and it is designed to contextualise the temporal relationship between the activity features. To obtain a probability distribution for the start and end predicted positions, we utilise two different fully connected layers to produce scores associated to the probabilities of each output of the GRU being the start/end of the location. Then, we take the softmax of these scores and thus obtain vectors $\hat{\boldsymbol{\tau}}^s, \hat{\boldsymbol{\tau}}^e \in \mathbb{R}^t$ containing a categorical probability distribution. Even though we do not constrain the starting and ending points to follow the right order in time, this does not result in any difficulties in practice.

5.4 Training

Our method is trained end-to-end on a dataset consisting of annotated tuples (V, Q, t^s, t^e). Note that each video V may include more than one moment and may therefore appear in multiple tuples. We treat each training sample independently. Given a new video and sentence tuple (V_r, Q_r), our model predicts the most likely temporal localization of the moment described by Q_r in terms of its start and end positions, $t_r^{s\star}$ and $t_r^{e\star}$, in the video. We use the Kullback-Leibler divergence and a spatial loss proposed in Chapter 4 of this thesis. We explain this in more detail in the supplemental material. Given the predicted/ground truth starting/ending times of the moment, we use the following loss function during training:

$$L_{\mathrm{KL}} = D_{\mathrm{KL}}(\hat{\boldsymbol{\tau}}^s \parallel \boldsymbol{\tau}^s) + D_{\mathrm{KL}}(\hat{\boldsymbol{\tau}}^e \parallel \boldsymbol{\tau}^e) \tag{5.15}$$

where D_{KL} is the Kullback-Leibler divergence. Moreover, inspired by our previous work, we use a spatial loss that aims to create activity features that are good at identifying where the action is occurring. This loss, equation 5.16, receives as input $\mathbf{y} = \mathrm{softmax}(g(\mathbf{A}))$ where $\mathbf{A}$ is the matrix that results by concatenating the improved activity representations over time, and g is a linear mapping that gives us a score for each activity representation. We apply a softmax function over these and our loss penalizes if this normalized score is large for those features associated to positions that lie outside the temporal location of the query.

$$L_{\mathrm{spatial}} = -\sum_{i=1}^{t}(1 - \delta_{\tau^s \leq i \leq \tau^e}) \log(1 - y^i) \tag{5.16}$$

where δ is the Kronecker delta. The final loss for training our method is the sum of the two individual losses defined previously setting $\mathcal{L} = L_{\mathrm{KL}} + L_{\mathrm{spatial}}$. During inference, we predict the starting and ending positions using the most likely locations given by the estimated distributions, using $\hat{\tau}^s = \mathrm{argmax}(\hat{\boldsymbol{\tau}}^s)$ and $\hat{\tau}^e = \mathrm{argmax}(\hat{\boldsymbol{\tau}}^e)$. Since values correspond to positions in the feature domain of the video, so we convert them back to time positions.

5.5 Experiments and Results

In this section we give details about the experiments performed to test our proposed approach. We begin by describing the datasets used in our study, to later provide details about the training and evaluation procedures we followed.

5.5.1 Datasets

To evaluate our proposed approach we work with three widely utilized and challenging datasets, namely Charades-STA [Gao et al., 2017a], ActivityNet Caption [Caba Heilbron et al., 2015; Krishna et al., 2017] and TACoS [Rohrbach et al., 2014], explained in detail in section 4.4. In addition to these, we also consider the YouCookII dataset [Zhou et al., 2018b,a]. This decision is motivated by its activity-centric nature as YoucookII is built upon instructional videos making it an excellent candidate to evaluate our proposals.

YouCookII: consists of 2,000 long untrimmed videos from 89 cooking recipes obtained from YouTube by Zhou et al. [2018b]. Each step for cooking these dishes was annotated with temporal boundaries and aligned with the corresponding section of the recipe. Recipes are written following the usual style of the domain [Lin et al.; Gerhardt et al., 2013], which includes very specific instruction-like statements with a wide degree of detail. The videos on this dataset are taped by individual persons at their houses while following the recipes using movable cameras. Similarly to TACoS, the average video length is 5.26 minutes. In terms of relevant moment segments, each video has 7.73 moments on average, with each segment being 19.63 seconds long on average. Videos have a minimum of three and a maximum of 16 moments.

In Table 5.1 we can see the summarize details of the exact sizes of the train/validation/test splits for each dataset.

Dataset	Train	Validation	Test
Charades-STA	12,408	3,720	-
ActivityNet Captions	37,414	17,502	-
YouCookII	10,337	3,492	-
TACoS	10,146	4,589	4,083

Table 5.1: Exact sizes of the train/validation/test splits for each dataset. Charades-STA, ActivityNet and YouCookII withheld the test set, therefore, the common practice is to report the data in the validation set

5.5.2 Implementation Details

We first pre-process the videos by extracting features of size 1024 using I3D with average pooling, taking as input the raw frames of dimension 256 × 256, at 25fps. We use the pre-trained model trained on Kinetics for TACoS, ActivityNet and YouCookII released by Carreira and Zisserman [2017]. For Charades-STA, we use the pre-trained model trained on Charades. We extract the top 15 object detections in terms of confidence for each of the keyframes using Faster-RCNN.

All of our models are trained in an end-to-end fashion using ADAM Kingma and Ba [2015] with a learning rate of 10^{-4} and weight decay 10^{-3}. As mentioned earlier, our temporal graph is modelled using a two-layer BiGRU. We use a hidden size of 256 and to prevent over-fitting, we add a dropout of 0.5 between the two layers.

5.5.3 Evaluation

We evaluate our model by two widely use metrics proposed by Gao et al. [2017a]. The Recall at various thresholds of the temporal Intersection over Union ($R@\alpha$) measuring the percentage of predictions that have tIoU with ground truth larger than certain α, and mean averaged tIoU (mIoU). We use three α threshold values 0.3, 0.5 and 0.7.

N	R@0.3	R@0.5	R@0.7	R@0.9	mIoU
0	47.46	22.88	14.38	6.00	33.67
1	73.21	55.32	36.02	11.48	52.11
2	79.01	67.16	48.71	17.97	59.30
3	**79.25**	**68.41**	**50.56**	**19.14**	**60.29**
4	70.99	60.31	44.16	17.32	54.01

Table 5.2: Performance when using a different number of iterations (N) for the message-passing algorithm, on a subsection of the training split of Charades-STA.

5.5.4 Ablation Study

To show the effectiveness of our proposals we perform several ablation studies each aimed at assessing the contribution of different components of our model. All of our ablative experiments are based on a segment of the training split of the Charades-STA dataset. As mIoU provides a more comprehensive evaluation of the performance of our model, we utilised this metric to select the best model configuration for the rest of the experiments in this paper.

Since feed-forwarding through our proposed spatial graph is an iterative process, we first studied the impact on the performance of the number of iterations (N) utilised in the message-passing algorithm of our full model. As shown in Table 5.2, we experimented setting N to a minimum value of 0 (where nodes are not updated at all) up to a maximum number of 4 iterations. As expected, performance tends to improve with larger values of N, with a saturation point at $N = 3$. Based on these results, all of our models in the rest of this paper are trained to utilise three iterations.

Table 5.3 summarises the results of our ablation studies, which include: (1) Concatenating the mean-pooling of the features extracted by Faster RCNN directly with the activity representation, therefore eliminating the human and object nodes (No Graph) to assess the relevance of our graph in using the spatial information. (2) Evaluating the importance of distinguishing between human versus object features by testing how our model performs when assigning all the detected features to one spatial node (No Node Types). In (3) and (4) we remove the use of human (No $\mathcal{H}$) and object (No $\mathcal{O}$) spatial information, respectively. (5) Assessing the contribution of the linguistic nodes (No LA) by modifying our graph so that it only contains a single textual node connected to the rest of the graph in a way analogous to our full model. (6) Testing the importance of the spatial loss L_{spatial} which encourages our model to focus on the features within the segment of interest. As can be seen, the importance of each one of our studied components is validated as ablations always result in consistent performance drops in terms of both mIoU as well as tIoU at the majority of α thresholds.

No Node Types This ablation experiment is intended to show the importance of considering the Faster-RCNN features related to human labels as a different source of information. The experiment consists of assigning the same 15 object features ex-

Model	R@0.3	R@0.5	R@0.7	R@0.9	mIoU
(1) No Graph	44.32	13.46	7.66	2.50	31.09
(2) No Node Types	74.78	61.24	43.35	14.59	55.18
(3) No $\mathcal{H}$ Node	75.46	60.60	43.31	15.51	55.32
(4) No $\mathcal{O}$ Node	75.66	61.28	44.08	15.39	56.13
(5) No LA	76.79	66.32	49.92	20.87	58.93
(6) No L_{spatial}	76.79	66.60	**52.54**	**23.57**	59.95
Full Model	**79.25**	**68.41**	50.56	19.14	**60.29**

Table 5.3: Results of our ablation studies, performed on a section of the training split of Charades-STA.

tracted for each of the keyframes only to the Object node $\mathcal{O}$. In this way, limit the ability of the network to only be able to find relations between objects and activity representations, but without reducing the total amount of data that is available to it. We consider this experiment is very relevant as it shows that the additional information provided by the objects detected is not the only reason to explain the performance improvements, but rather the way in which this data is used is more relevant. In fact, enabling the model to obtain state-of-the-art performance in different and challenging benchmarks.

No Language Attention In this case, we replace the set of linguistic nodes by a single query node $\mathcal{Q}$. It receives a high-dimensional representation (denoted by q) of the natural language query Q, as can be seen in Figure 5.2. This high-dimensional representation is constructed using a function $F_Q : Q \mapsto q$ that first maps each word w_j for $j = 1, \ldots, m$ in the query to a semantic embedding vector $h_j \in \mathbb{R}^{d_w}$, where d_w defines the hidden dimension of the word embedding. Representations for each word are then aggregated using mean pooling to get a semantically rich representation of the whole query.

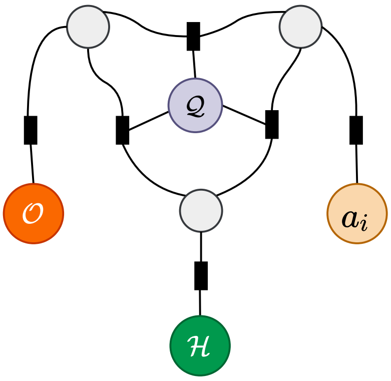


Figure 5.2: Spatial graph with a single query node $\mathcal{Q}$

Although the query node is generic, in this work, we use a bi-directional GRU

[Cho et al., 2014] on top of GLoVe word embeddings, which are pre-trained on a large collection of documents, for computing the h_j. Therefore, our query functions F_Q is parameterised by both GLoVe embedding and the GRU.

Again we capture the relationship between this high-dimensional representation of the query and any observation of the nodes human $\mathcal{H}$, object $\mathcal{O}$ and activity $\mathcal{A}$, using a linear mapping function f specific for each node, as follows:

$$\Phi^{n}_{\mathcal{Q},\mathcal{A}} = f_{\mathcal{Q},\mathcal{A}}(q, a^{n}) \tag{5.17}$$

$$\Phi^{j,n}_{\mathcal{Q},\mathcal{O}} = f_{\mathcal{Q},\mathcal{O}}(q, o^{j,n}) \tag{5.18}$$

$$\Phi^{k,n}_{\mathcal{Q},\mathcal{H}} = f_{\mathcal{Q},\mathcal{H}}(q, h^{k,n}) \tag{5.19}$$

where functions $f_{\mathcal{Q},\mathcal{A}}, f_{\mathcal{Q},\mathcal{O}}, f_{\mathcal{Q},\mathcal{H}}$ are simple linear projections. For example, in the case of the object observations we have $f_{\mathcal{Q},\mathcal{O}}(q, o^{j,n}) = W_{qo}[q; o^{j,n}] + b_{qo}$, where the subindex qo denotes the dependency of the parameters of the linear function which are specific for each relation. To compute the messages that are passed between the nodes, we utilize the following functions:

$$\Psi^{j,n}_{\mathcal{H},\mathcal{Q},\mathcal{O}} = f_{\mathcal{H},\mathcal{Q},\mathcal{O}}(\Phi^{j,n}_{\mathcal{Q},\mathcal{O}}, \textstyle\sum_{k=1}^{K} \Phi^{k,n}_{\mathcal{Q},\mathcal{H}}) \tag{5.20}$$

$$\Psi^{j,n}_{\mathcal{A},\mathcal{Q},\mathcal{O}} = f_{\mathcal{A},\mathcal{Q},\mathcal{O}}(\Phi^{j,n}_{\mathcal{Q},\mathcal{O}}, \Phi^{n}_{\mathcal{Q},\mathcal{A}}) \tag{5.21}$$

$$\Psi^{n}_{\mathcal{H},\mathcal{Q},\mathcal{A}} = f_{\mathcal{H},\mathcal{Q},\mathcal{A}}(\Phi^{n}_{\mathcal{Q},\mathcal{A}}, \textstyle\sum_{k=1}^{K} \Phi^{k,n}_{\mathcal{Q},\mathcal{H}}) \tag{5.22}$$

$$\Psi^{n}_{\mathcal{O},\mathcal{Q},\mathcal{A}} = f_{\mathcal{O},\mathcal{Q},\mathcal{A}}(\Phi^{n}_{\mathcal{Q},\mathcal{A}}, \textstyle\sum_{j=1}^{J} \Phi^{j,n}_{\mathcal{Q},\mathcal{O}}) \tag{5.23}$$

$$\Psi^{k,n}_{\mathcal{O},\mathcal{Q},\mathcal{H}} = f_{\mathcal{O},\mathcal{Q},\mathcal{H}}(\Phi^{k,n}_{\mathcal{Q},\mathcal{H}}, \textstyle\sum_{j=1}^{J} \Phi^{j,n}_{\mathcal{Q},\mathcal{O}}) \tag{5.24}$$

$$\Psi^{k,n}_{\mathcal{A},\mathcal{Q},\mathcal{H}} = f_{\mathcal{A},\mathcal{Q},\mathcal{H}}(\Phi^{k,n}_{\mathcal{Q},\mathcal{H}}, \Phi^{n}_{\mathcal{Q},\mathcal{A}}) \tag{5.25}$$

where again $f_{\mathcal{H},\mathcal{Q},\mathcal{O}}, f_{\mathcal{A},\mathcal{Q},\mathcal{O}}, f_{\mathcal{H},\mathcal{Q},\mathcal{A}}, f_{\mathcal{O},\mathcal{Q},\mathcal{A}}, f_{\mathcal{O},\mathcal{Q},\mathcal{H}}$ and $f_{\mathcal{A},\mathcal{Q},\mathcal{H}}$ are linear mappings, each receiving as input a concatenations of the corresponding features capturing. Finally, we update the representation of the human, action and object nodes based on the following formulas.

$$o^{j,n+1} = \sigma(m_o(\Psi^{j,n}_{\mathcal{H},\mathcal{Q},\mathcal{O}} \odot \Psi^{j,n}_{\mathcal{A},\mathcal{Q},\mathcal{O}}) \odot o^{j,0}) \tag{5.26}$$

$$a^{n+1} = \sigma(m_a(\Psi^{n}_{\mathcal{H},\mathcal{Q},\mathcal{A}} \odot \Psi^{n}_{\mathcal{O},\mathcal{Q},\mathcal{A}}) \odot a^{0}) \tag{5.27}$$

$$h^{k,n+1} = \sigma(m_h(\Psi^{k,n}_{\mathcal{O},\mathcal{Q},\mathcal{H}} \odot \Psi^{k,n}_{\mathcal{A},\mathcal{Q},\mathcal{H}}) \odot h^{k,0}) \tag{5.28}$$

where $\odot$ is the element-wise product and m_o, m_a, m_h are again linear functions.

5.5.5 Comparison with the state-of-the-art

We compare the performance of our proposed approach on the datasets considered against several prior works. We consider a broad selection of models based on different approaches, specifically proposal-based techniques including CTRL [Gao et al., 2017a], SAP [Chen and Jiang, 2019], MAN [Zhang et al., 2019] and CBP [Wang et al., 2020], as well as TripNet [Hahn et al., 2019], a method based on reinforcement learn-

Method	R@0.3	R@0.5	R@0.7	mIoU
Random	-	8.51	3.03	-
CTRL	-	21.42	7.15	-
ABLR †	-	24.36	9.00	-
TripNet	51.33	36.61	14.50	-
CBP	50.19	36.80	18.87	35.74
MAN	-	46.53	22.72	-
EXCL	65.10	44.10	22.60	-
DRN	-	53.09	31.75	-
TMLGA	67.53	52.02	33.74	48.22
LGVTI	72.96	59.46	35.48	51.38
Ours	**73.36**	**59.62**	**41.24**	**53.62**

Table 5.4: Performance comparison of our spatio-temporal graph approach with existing methods for different tIoU α levels. Values are reported on the validation split of Charades-STA. † Results for ABLR are as reported by Chen and Jiang [2019].

Method	R@0.3	R@0.5	R@0.7	mIoU
Random	5.60	2.50	0.80	
CTRL	28.70	14.00	-	20.54
ABLR †	55.67	36.79	-	36.99
TripNet	48.42	32.19	13.93	-
CBP	54.30	35.76	17.80	36.85
TMLGA	51.28	33.04	19.26	37.78
LGVTI	**58.52**	**41.51**	23.07	41.13
Ours	57.89	41.49	**26.41**	**42.78**

Table 5.5: Performance comparison of our spatio-temporal graph approach with existing methods for different tIoU α levels. Values are reported on the validation split of ANet-Cap.

ing. In addition to that, we also compare our approach to more recent methods that do not rely on proposals, including ABLR [Yuan et al., 2019], ExCL [Ghosh et al., 2019], Our method, Chapter 4, TMLGA [Rodriguez-Opazo et al., 2020] and LGVTI [Mun et al., 2020]. Finally, we also consider a random baseline that simply selects an arbitrary video segment as the moment for each example.

Tables 5.4, 5.5 and 5.6 summarizes our results on the datasets Charades-STA, ActivityNet Captions and TACoS respectively, while also comparing the obtained performance to relevant prior work. It is possible to see that our method is able to outperform previous work by a consistent margin, especially for the $\alpha = 0.7$ band and also in terms of the mean tIoU (mIoU). Comparing results across these datasets, we also see that the performance of all models drops substantially on ActivityNet Captions and TACoS, compared to Charades-STA. We think this is mainly due to the

Method	R@0.3	R@0.5	R@0.7	mIoU
Random	1.81	0.83		-
CTRL	18.90	13.30	-	-
ABLR †	18.90	9.30	-	-
TripNet	23.95	19.17	9.52	-
CBP	27.31	24.79	19.10	21.59
EXCL	**44.20**	28.00	14.60	-
TMLGA	24.54	21.65	16.46	22.06
DRN	-	23.17	-	-
Ours	31.80	**28.69**	**24.91**	**26.42**

Table 5.6: Performance comparison of our spatio-temporal graph approach with existing methods for different tIoU α levels. Values are reported on the test split of TACoS.

Method	R@0.3	R@0.5	R@0.7	mIoU
Random	4.84	1.72	0.60	-
TMLGA	33.48	20.65	10.94	23.07
Ours	**43.73**	**29.93**	**17.61**	**30.43**

Table 5.7: Performance on **YouCookII** for different tIoU α levels.

nature of the datasets. On one side, TACoS contains a considerable amount of video moments that only span a few seconds, which is equivalent to two or three activity features. On the other side, ActivityNet Captions mostly has one query describing the complete extent of a video. This bias is hindering the training performance of proposal-free approaches. Moreover, ActivityNet Captions has queries that describe a video's intrinsic information, i.e., opening sequence or credits, where our method could not distinguish the temporal difference, and better temporal reasoning is necessary.

Table 5.7 shows the performance of our model on the YouCookII dataset. To the best of our knowledge, since no previous work has been evaluated on this dataset, we used our method [Rodriguez-Opazo et al., 2020] explained in Chapter 4 released by its authors as an additional baseline. As can be seen, we are able to outperform both the random baseline and TMLGA by a large margin, especially on the lower α bands.

Finally, we also study the effect of using a different pre-trained model to obtain activity representations in our proposed approach. Concretely, we test the performance of our model using VGG-16 features instead of I3D on Charades-STA. Table 5.8 summarises our obtained results and compares them to prior work also utilising these features. As can be seen, although VGG features provide lower performance than I3D in our experiments, therefore experimentally validating our choice, our model is still able to outperform existing approaches also using these features by a large

Model	R@0.3	R@0.5	R@0.7	R@0.9	mIoU
SAP	-	27.42	13.36	-	-
MAN	-	41.24	20.54	-	-
DORi	61.83	43.47	26.37	7.63	42.52

Table 5.8: Performance of our spatio-temporal graph approach with VGG-16 features, We compared to relevant prior work that uses the same type of features.

margin, showing the superiority of our proposed approach.

5.5.6 Qualitative results

Language Attention In the following Figures 5.3 and 5.4, we present a set of samples of the multihead attention to the query sentence on the Charades-STA dataset. In the case of node $\mathcal{SN}$ which models the subject-object relationship focuses on the dot at the end of the sentence, which can be interpreted as the need of the node to read the whole sentence. However, node $\mathcal{VN}$ consistently focuses on the verb of the sentences, and node $\mathcal{SV}$ focuses on the verb and objects in the scene.

Visualization of localization

In the following Figures, we present success and failure cases of our method on Charades-STA, YouCookII and TACoS dataset. Each visualisation is showing a subsample of the keyframes inside of the prediction with their corresponding spatial observations. In green observations associated with the human node $\mathcal{H}$ and orange for the object node $\mathcal{O}$. Moreover, each visualisation is presenting the ground-truth and predicted localisation in seconds of the given query.

5.5.6.1 Charades-STA

Success cases of our algorithm on the Charades-STA dataset can be seen in Figure 5.5. In Figure 5.5a, given the query "a person cooks a sandwich on a panini maker" our method could localize the moment at a tIoU of 99.56%. The label of the features extracted by Faster-RCNN to localize the query are *'bottle', 'counter', 'door', 'drawer', 'faucet', 'floor', 'glasses', 'hair', 'jacket', 'jeans', 'kitchen', 'microwave', 'pants', 'shelf', 'shirt', 'sink', 'stove', 'sweater',* ***'toaster'****, 'wall', 'window', 'woman'.*

In the case of Figure 5.5b, given the query "the person closes a cupboard door." our method could localize the moment at a tIoU of 97.88%. The features extracted by Faster RCNN for this query are *'arm', 'building',* ***'cabinet'****, 'counter', 'door', 'faucet', 'hair', 'hand', 'head', 'jacket', 'kitchen', 'man', 'microwave', 'refrigerator', 'shirt', 'sink', 'sleeve', 'stove', 'sweater', 'wall', 'window', 'woman'.*

Failure cases of our method are presented in Figure 5.6. In the first example, given a query "a person opens a door goes into a room." our method could detect correct spatial features, such as 'door' and 'knob', and the correct span of the query,

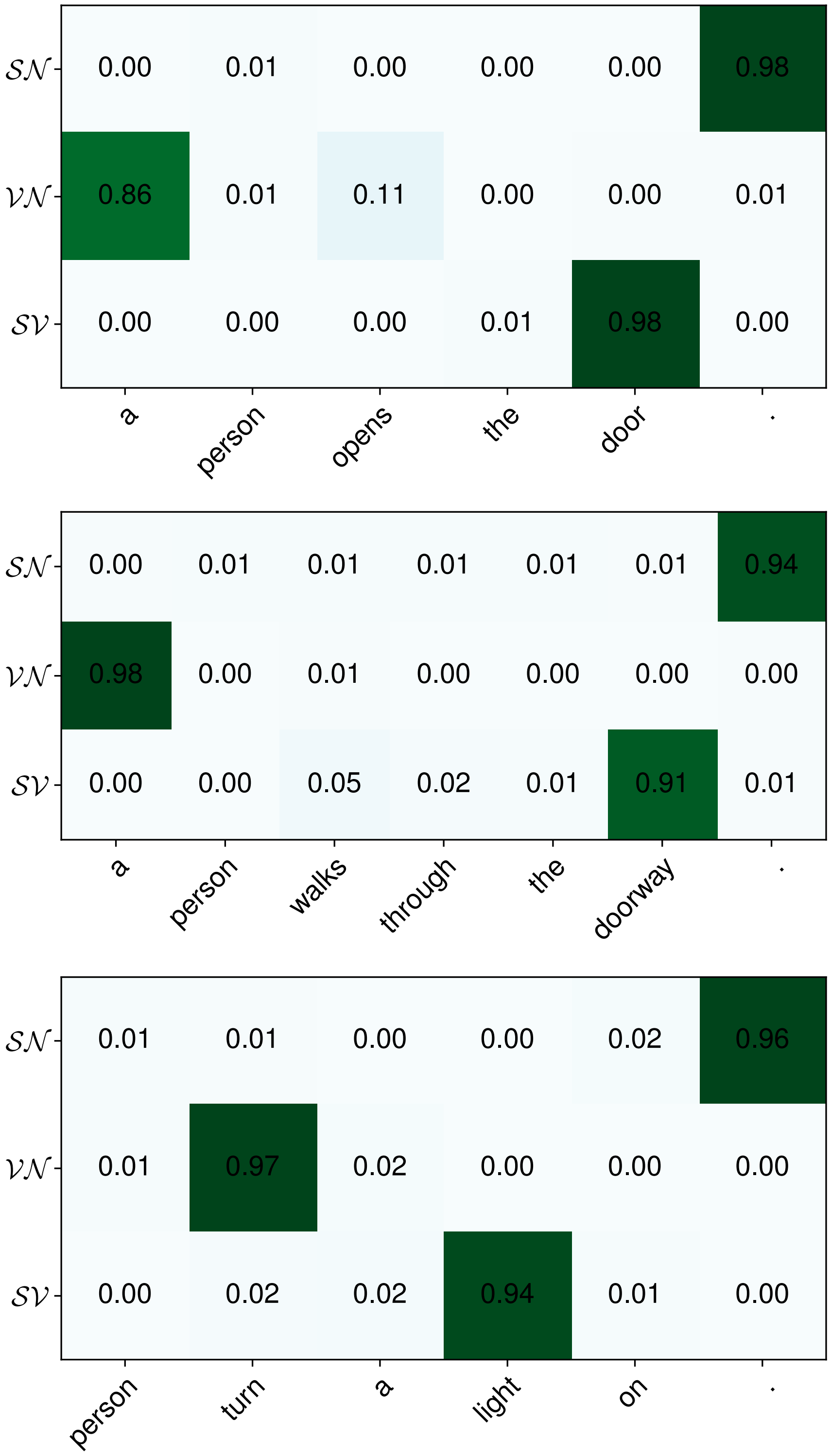


Figure 5.3: First example of the linguistic nodes attentions on Charades-STA.

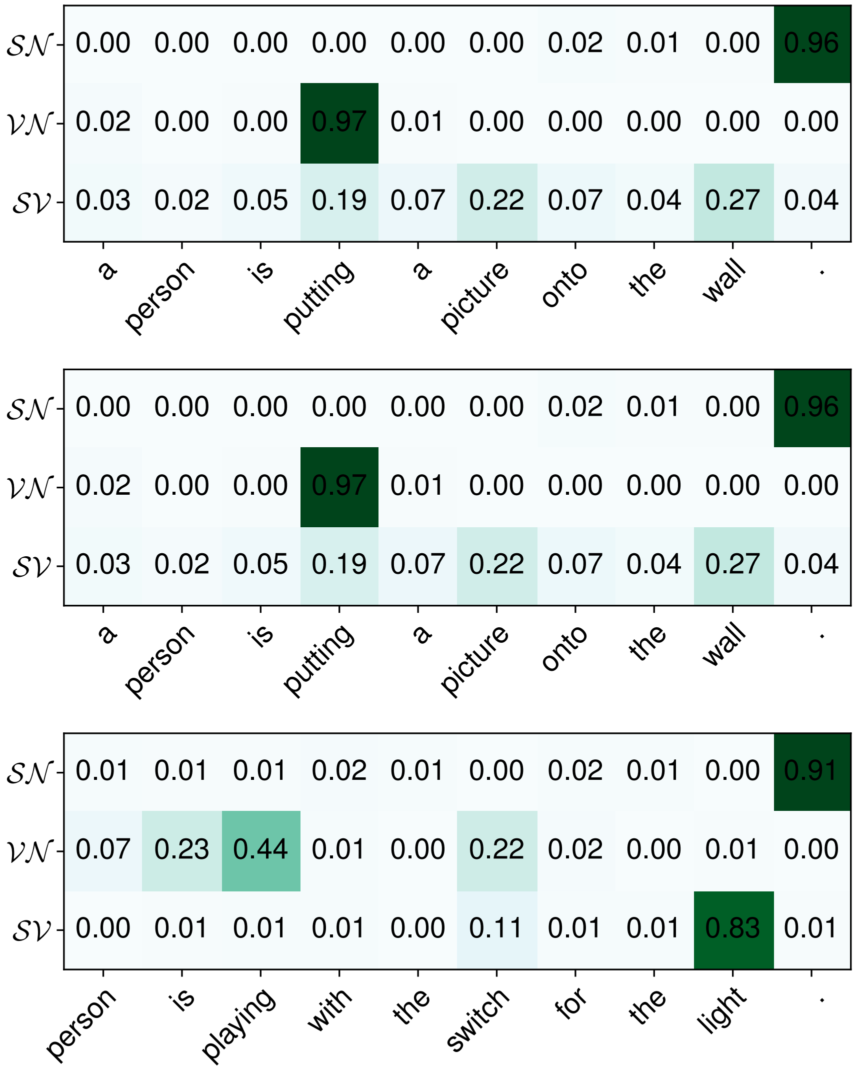
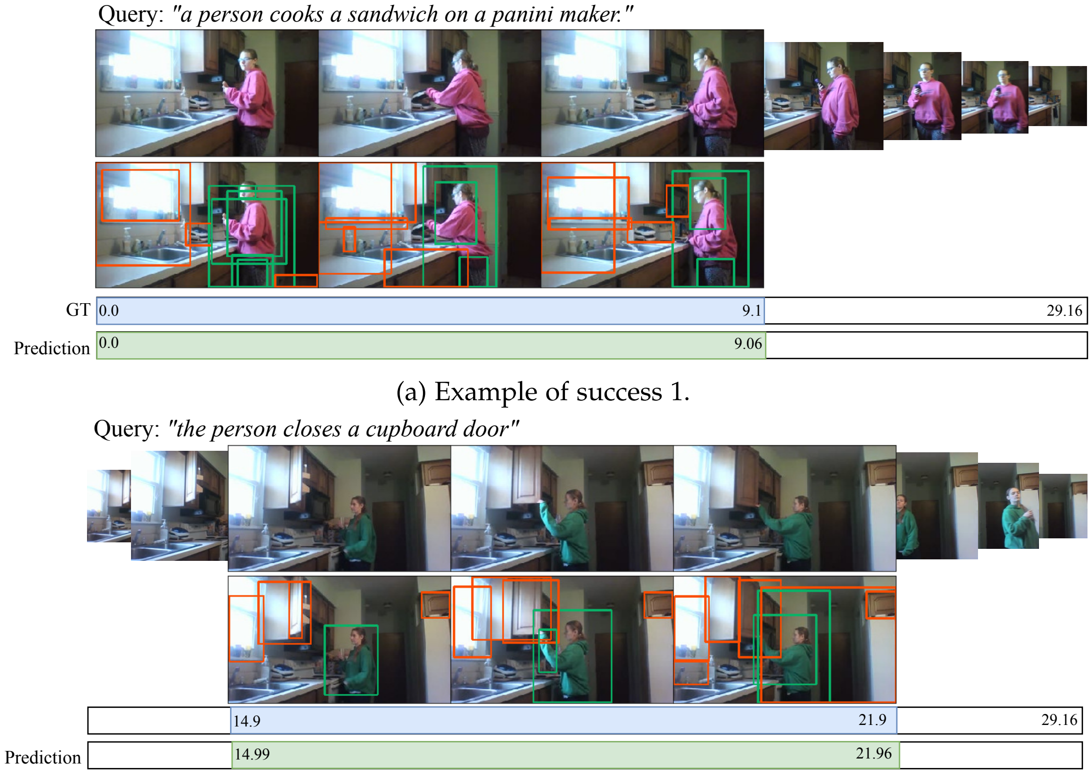


Figure 5.4: Second example of the linguistic nodes attentions on Charades-STA.

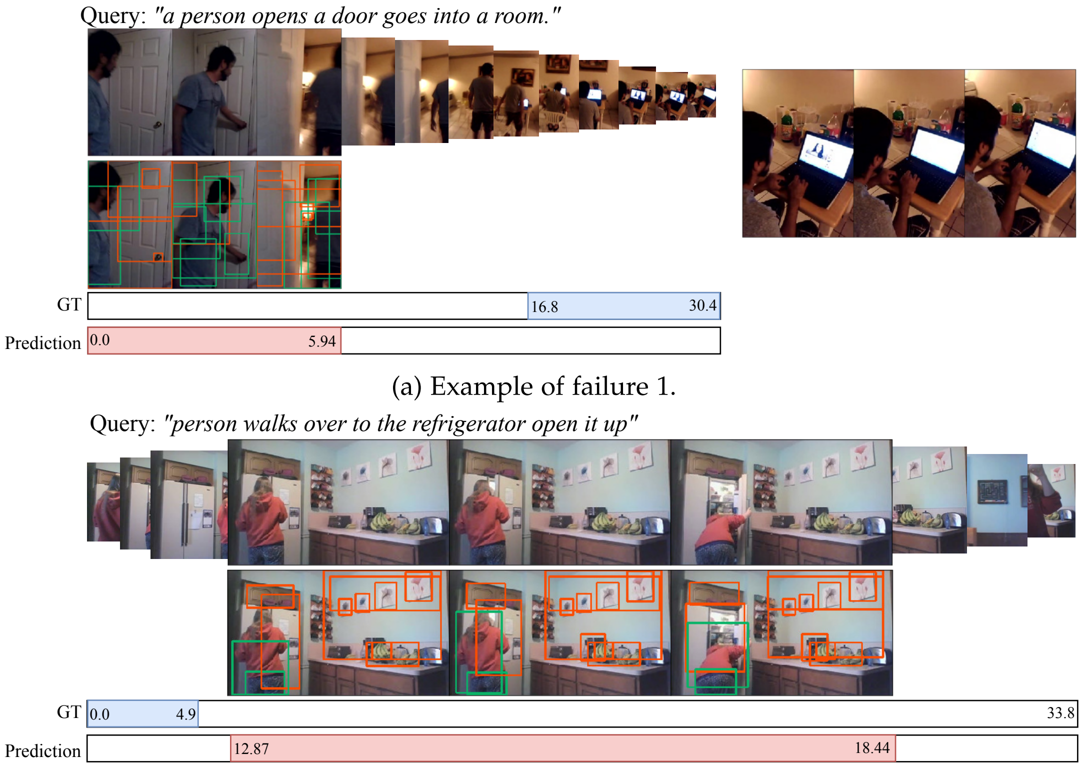
according to our qualitative evaluation. However, in this case, the annotation for the query is localised incorrectly in the video. It refers to the last part of the video, where a person is using a laptop, as can be seen at the right of Figure 5.6a. In Fig. 5.6b we can see our method localising the query "person walks over to the refrigerator open it up", however, the annotation is not considering that the moment is performed two times in the video.


Query: *"a person cooks a sandwich on a panini maker."*
GT
0.0
9.1
29.16
Prediction
0.0
9.06
Query: *"the person closes a cupboard door"*
14.9
21.9
29.16
Prediction
14.99
21.96

(a) Example of success 1.

(b) Example of success 2.

Figure 5.5: Success examples of our spatio-temporal graph approach on Charades-STA dataset.


Query: "a person opens a door goes into a room."
GT
16.8
30.4
Prediction
0.0
5.94
Query: "person walks over to the refrigerator open it up"
GT
0.0
4.9
33.8
Prediction
12.87
18.44

(a) Example of failure 1.

(b) Example of failure 2.

Figure 5.6: Failure examples of our spatio-temporal graph approach on Charades-STA.

5.5.6.2 YouCookII

Although videos in YouCookII are much longer than videos in Charades-STA, our method still can get good localisation performance. In Figure 5.8a given the query "spread the sauce onto the dough" our method localise the query at a tIoU of 98.57%. The label of the feature extracted by Faster-RCNN on this case are *'bacon', 'bird', 'board', 'bottle', 'bowl', 'cabinet', 'cake', 'cherry', 'chocolate', 'cookie', 'counter', 'cutting board', 'dessert', 'door', 'drawer', 'finger', 'floor', 'fork', 'fruit', 'glass', 'grape', 'ground', 'hand', 'handle', 'jeans', 'ketchup', 'knife', 'meat', 'olive', 'pancakes', 'pepperoni', 'person', 'phone',* ***'pizza'***, *'plant', 'plate',* ***'sauce'***, *'saucer', 'shirt', 'sleeve', 'spoon', 'table', 'towel', 'tree', 'wall'*.

Figure 5.8 presents a success case of our method on YouCookII dataset. The visualization is presenting a subsample of the key-frames inside of the prediction with their corresponding spatial observations, with green observations associated with the human node $\mathcal{H}$ and orange to the object node $\mathcal{O}$. Moreover, each visualization is presenting the ground-truth localisation and predicted localisation of the given query. As shown in Figure 5.8, given the query "cover the dish with mashed potatoes", our method could localise the moment at a tIoU of 98.88%. The most relevant features extracted by Faster-RCNN to localise the query are *'arm', 'bowl', 'cake', 'hand', 'kitchen', 'man', 'mug', 'spoon', 'stove', 'tray'*.

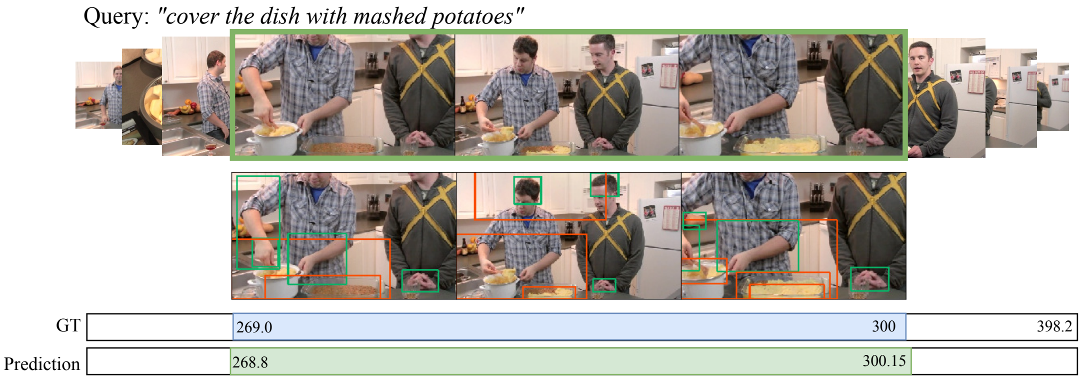
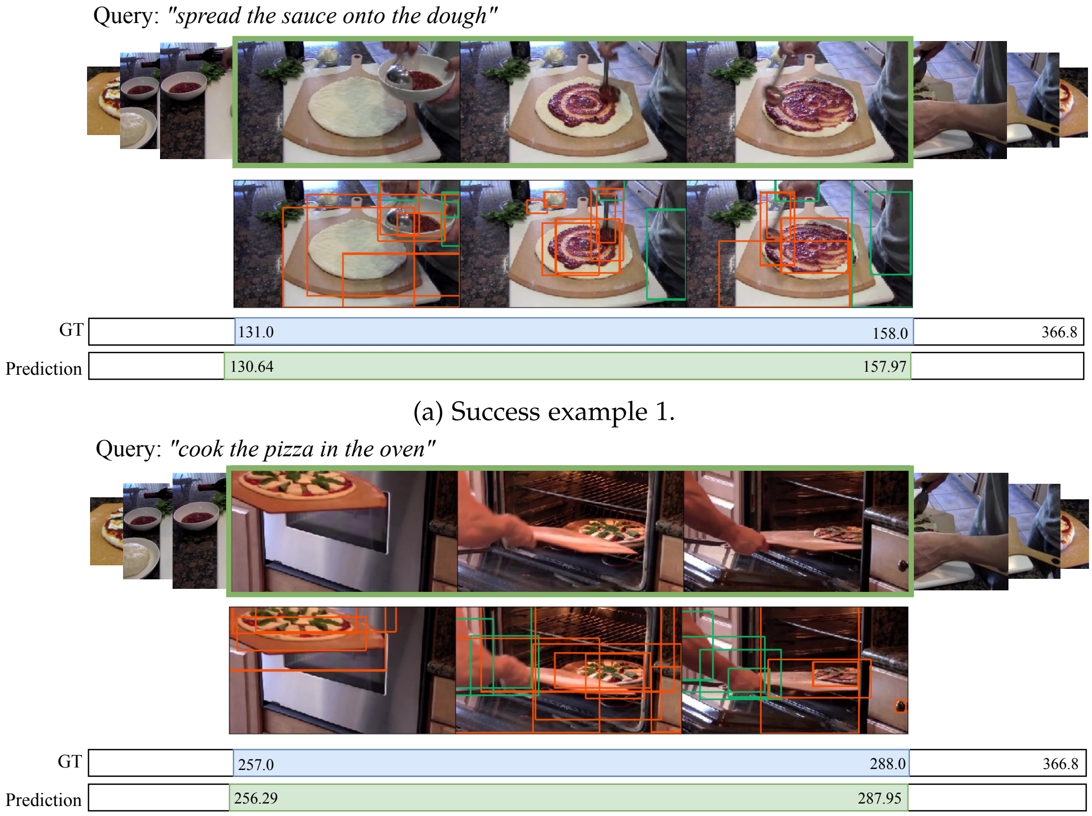


Figure 5.7: Visualization of a success case of our method in the YouCookII dataset. The second row shows the observations associated to the Human node (green) and Object node (orange).

Figure 5.8b shows the query "cook the pizza in the oven", which belong to the same video. In this case, the label of the features extracted by Faster-RCNN are *'arm', 'bar', 'board', 'building', 'cabinet', 'car', 'ceiling', 'cheese', 'cord', 'counter', 'crust', 'cucumber', 'curtain', 'door', 'drawer', 'fireplace', 'floor', 'food', 'fork', 'glass', 'grill', 'hand', 'hotdog', 'key', 'keyboard', 'kitchen', 'knife', 'knob', 'laptop', 'leaf', 'leaves', 'leg', 'light', 'man', 'microwave', 'mouse',* ***'oven'***, *'oven door', 'person',* ***'pizza'***, *'plate', 'pole', 'rack', 'roof', 'room', 'salad', 'screen', 'shadow', 'sleeve', 'slice', 'spinach', 'stove', 'table', 'television', 'thumb', 'tracks', 'train', 'tray', 'vegetable', 'vegetables', 'wall', 'window', 'wood'* and our method could localise the query with a temporal intersection over union of 97.60%.

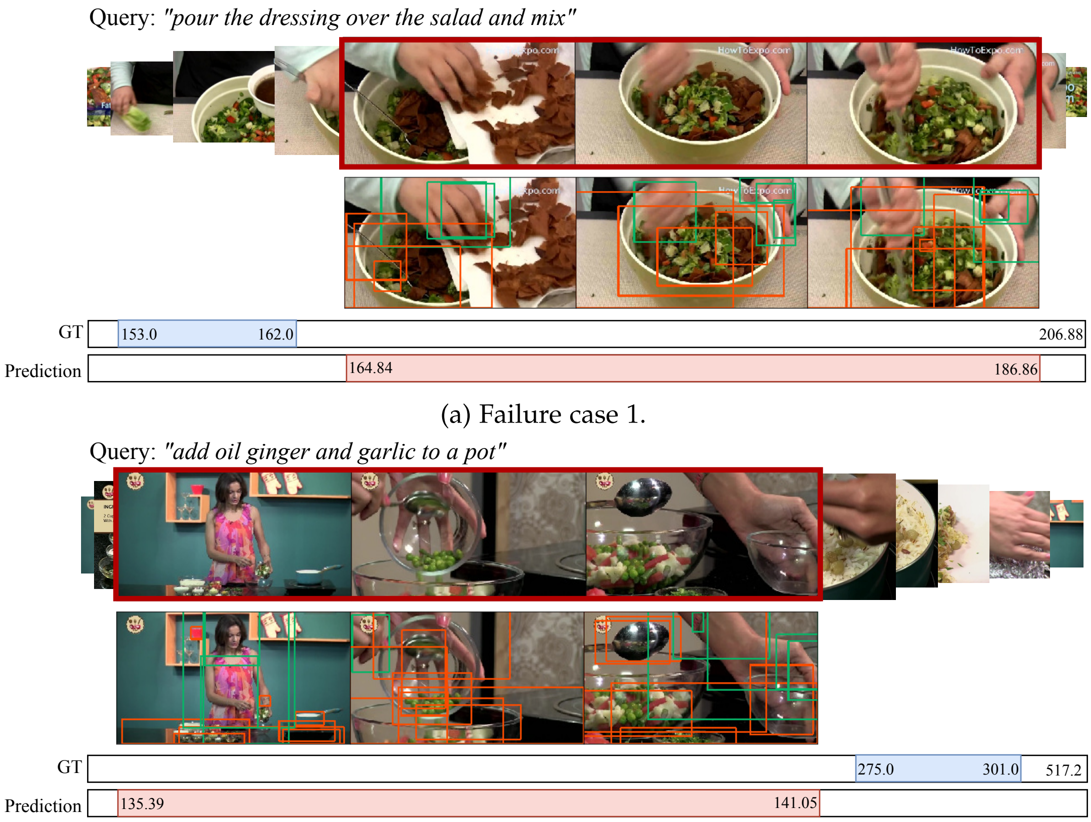


(a) Success example 1.

(b) Success example 2.

Figure 5.8: Success examples of our spatio-temporal graph approach in the YouCookII dataset.

Failure cases of our method on YouCookII dataset are presented in Figure 5.9. In these cases, it is possible to see that our approach is able to recognise the activity *add* and *mix* correctly. However, the objects "dressing, ginger and garlic" are not detected by Faster-RCNN, probably given that the object detector has not been trained to deal with some of the kinds of objects present on this dataset. We think this naturally hinders the disambiguation capabilities of our model, especially in terms of the repetitive actions such as adding, mixing and pouring, which are often performed throughout recipes like the one depicted in the example.


Query: "pour the dressing over the salad and mix"
GT
153.0
162.0
206.88
Prediction
164.84
186.86
Query: "add oil ginger and garlic to a pot"
GT
275.0
301.0
517.2
Prediction
135.39
141.05

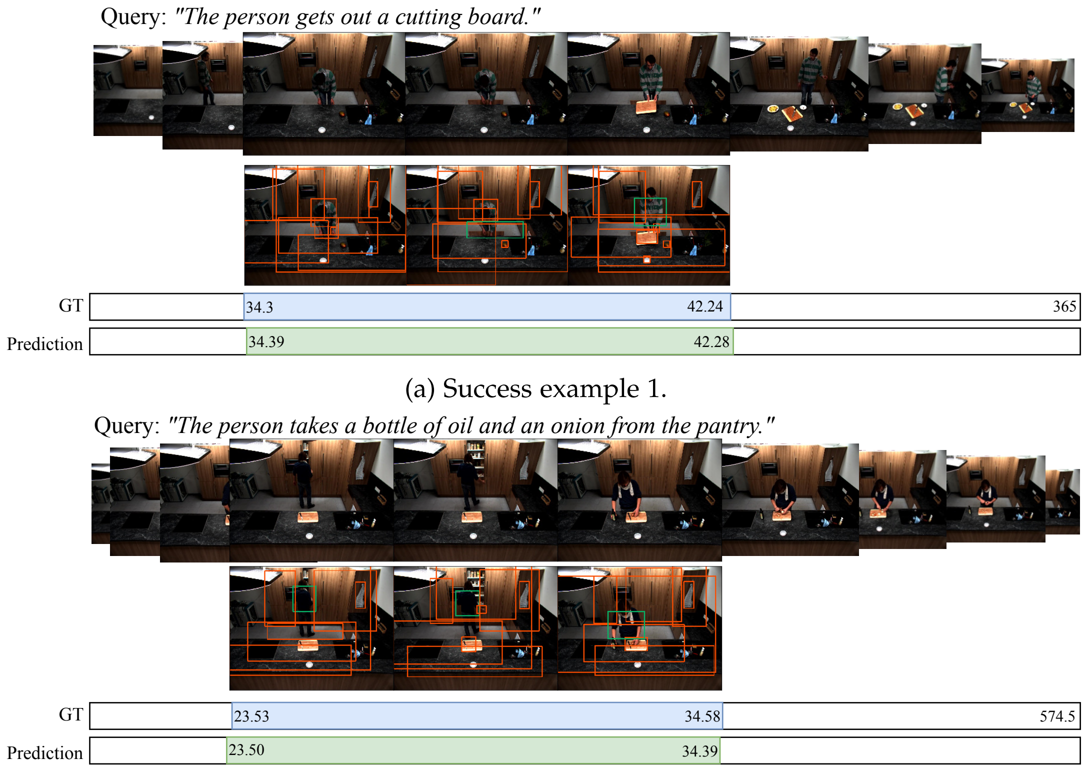
(a) Failure case 1.

(b) Failure case 2.

Figure 5.9: Failure cases of our spatio-temporal graph approach in the YouCookII dataset.

5.5.6.3 TACoS

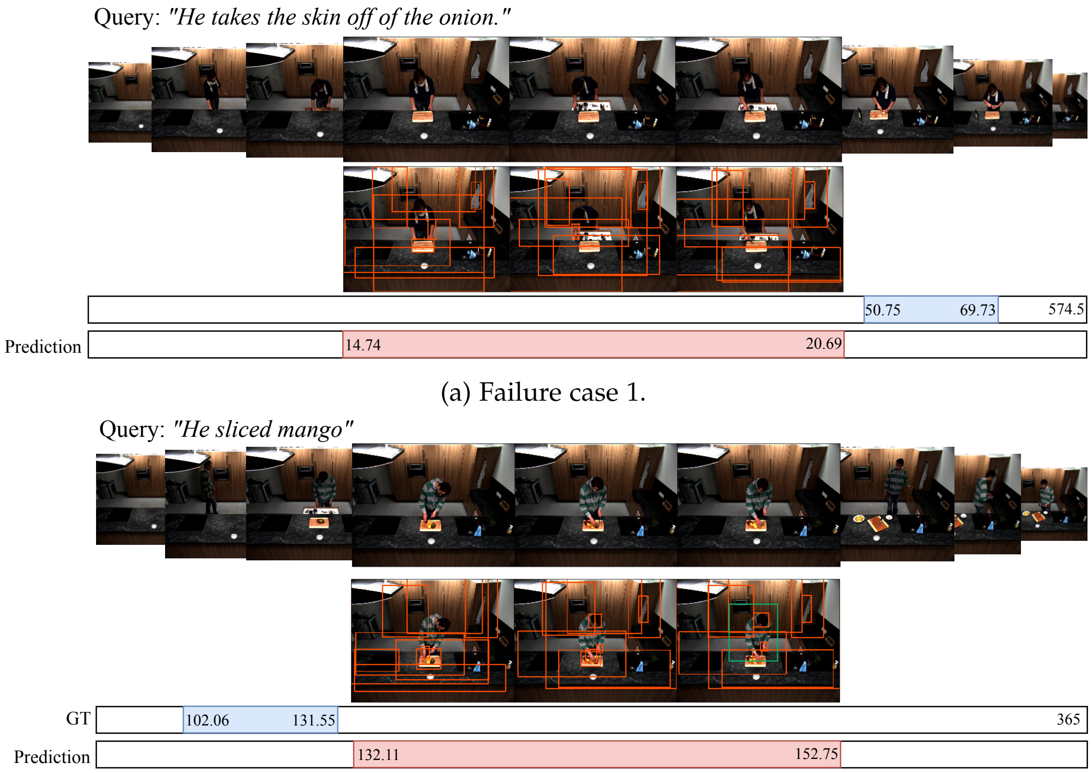
Figures 5.10 and 5.11 show two examples of success and failure cases on the TaCoS dataset, respectively. It is possible to see the how challenging this dataset is in general, as in the the cases where our approach fails it is in fact difficult even for us to localise the given query.


Query: *"The person gets out a cutting board."*
GT
34.3
42.24
365
Prediction
34.39
42.28
Query: *"The person takes a bottle of oil and an onion from the pantry."*
GT
23.53
34.58
574.5
Prediction
23.50
34.39

(a) Success example 1.

(b) Success example 2.

Figure 5.10: Success examples of our spatio-temporal graph approach in the TACoS dataset.



(a) Failure case 1.

(b) Failure case 2.

Figure 5.11: Failure cases of our spatio-temporal graph approach in the TACoS dataset.

5.6 Summary

In this chapter, we have presented a novel approach to temporal moment localisation of a language query in a video. Our approach consists of a spatio-temporal graph for capturing the relationships between detected human-human, human-objects and activities over time. We proposed a message-passing algorithm that propagates information across the graph conditioned on a natural language query, to ultimately infer the arbitrarily long segment in the video that most likely described by the query. Using our approach, we are able to achieve state-of-the-art results on several benchmark datasets.

Chapter 6

A Multi-Modal Approach to Fine-Grained Opinion Mining on Video Reviews

This chapter is based on:

Marrese-Taylor, E.*;**Rodriguez-Opazo, C.***; Balazs-Thenot J.; Gould, S; Matsuo Y. A Multi-modal Approach to Fine-grained Opinion Mining on Video Reviews *Proceedings of the 2nd Grand Challenge and Workshop on Human Multimodal Language* part of the 56th Annual Meeting of the Association for Computational Linguistics, 2020.

In this chapter, we study the problem of fine-grained opinion mining in video reviews. In contrast with the previous chapters in this thesis, we made use of all the available modalities present in the videos —audio, vision and language— to analyze its information. Specifically, we refer to language for the transcript of the speech made within the video. The well-known setting of opinion mining refers to the task of automatically extracting the author's opinion from texts. Despite the recent advances in opinion mining for written reviews, few works have tackled other sources of reviews such as videos. We argue that video reviews have become indispensable to people's decision-making process of buying.

In light of this issue, we propose a multi-modal approach for mining fine-grained opinions from video reviews that can determine the aspects of the item under review discussed in the video and the sentiment orientation towards them. Our approach works at the sentence level without the need for time annotations and uses features derived from the audio, video and language transcriptions. We evaluate our approach on two datasets and show that leveraging the video and audio modalities provides increased performance over text-only baselines. Our method provides evidence that these additional modalities are key in better understanding video reviews.

* Equal contribution

6.1 Introduction

Sentiment analysis (SA) is an important task in natural language processing, aiming at extracting opinions and identifying the emotions and subjectivity express in the opinion. As a result, sentiment can be automatically collected, analyzed and summarized. Because of this, SA has received much attention not only in academia but also in industry, helping provide feedback based on customers' opinions about products or services. The underlying assumption in SA is that the entire input has an overall polarity, however, this is usually not the case. For example, laptop reviews generally not only express the overall sentiment about a specific model (e.g., "This is a great laptop"), but also relate to its specific aspects, such as the hardware, software or price. Subsequently, a review may convey opposing sentiments (e.g., "Its performance is ideal, I wish I could say the same about the price") or objective information (e.g., "This one still has the CD slot") for different aspects of an entity. Aspect-based sentiment analysis (ABSA) or fine-grained opinion mining aims to extract opinion targets or aspects of entities being reviewed in a text, and to determine the sentiment reviewers express for each. ABSA allows us to evaluate aggregated sentiments for each aspect of a given product or service and gain a more granular understanding of their quality. This is of especial interest for companies as it enables them to refine specifications for a given product or service, and leading to an improved overall customer satisfaction.

Fine-grained opinion mining is also important for a variety of NLP tasks, including opinion-oriented question answering and opinion summarization. In practical terms, the ABSA task can be divided into two sub-steps, namely aspect extraction (*AE*) and (aspect level) sentiment classification (*SC*), which can be tackled in a pipeline fashion, or simultaneously (*AESC*). These tasks can be regarded as a token-level sequence labeling problem, and are generally tackled using supervised learning. The 2014 and 2015 SemEval workshops, co-located with COLING 2014 and NAACL 2015 respectively, included shared tasks on ABSA [Pontiki et al., 2014] and also followed this approach, which has also served as a way to encourage developments alongside this line of research [Mitchell et al., 2013; Irsoy and Cardie, 2014; Liu et al., 2015; Zhang et al., 2015].

The flexibility provided by the deep learning setting has helped multi-modal approaches to bloom. Examples of this include tasks such as machine translation [Specia et al., 2016; Elliott et al., 2017], word sense disambiguation [Chen et al., 2015], visual question answering [Chen et al., 2017], language grounding [Beinborn et al.; Lazaridou et al., 2015], and sentiment analysis [Poria et al., 2015; Zadeh et al., 2016]. Specifically in this last example, the task focuses on generalizing text-based sentiment analysis to opinionated videos, where three communicative modalities are present: language (spoken words), visual (gestures), and acoustic (voice).

Although reviews often come in the form of a written commentary, people are increasingly turning to video platforms such as YouTube looking for product reviews to help them shop. In this context, Marrese-Taylor et al. [2017] explored a new direction, arguing that video reviews are the natural evolution of written product reviews and introduced a dataset of annotated video product review transcripts. Similarly, Garcia et al. [2019b] recently presented an improved version of the Persuasive Opinion Multimedia (POM) movie review dataset [Park et al., 2014], with annotated fine-grained opinions.

Although the videos in these kinds of datasets represent a rich multi-modal source of opinions, the features of the language in them may fundamentally differ from written reviews given that information is conveyed through multiple channels (one for speech, one for gestures, one for facial expressions, one for vocal inflections, etc.) In these, different information channels complement each other to maximize the coherence and clarity of their message. This means that although the content of each channel may be comprehended in isolation, in theory we need to process the information in all the channels simultaneously to fully comprehend the message [Hasan et al., 2019]. In this context, information extracted from nonverbal language in videos, such as gestures and facial expressions, as well as from audio in the manner of voice inflections or pauses, and from scenes, object or images in the video, become critical for performing well.

In light of this, our paper introduces a multi-modal approach for fine-grained opinion mining. We conduct extensive experiments on two datasets built upon transcriptions of video reviews, Youtubean [Marrese-Taylor et al., 2017] and a fine-grain annotated version of the POM dataset [Park et al., 2014; Garcia et al., 2019b], adapting them to our setting by associating timestamps to each annotated sentence using the video subtitles. Our results demonstrate the effectiveness of our proposed approach and show that by leveraging the additional modalities we can consistently obtain better performance.

6.2 Related Work

Our work is related to aspect extraction using deep learning, a task that is often tackled as a sequence labeling problem. In particular, our work is related to Irsoy and Cardie [2014], who pioneered in the field by using multi-layered RNNs. Later, Liu et al. [2015] successfully adapted the architectures by Mesnil et al. [2013] which were originally developed for slot-filling in the context of Natural Language Understanding.

Literature offers related work on the usage of RNNs for open domain targeted sentiment [Mitchell et al., 2013], where Zhang et al. [2015] experimented with neural CRF models using various RNN architectures on a dataset of informal language from Twitter.

Regarding target-based sentiment analysis, the literature contains several ad-hoc models that account for the sentence structure and the position of the aspect on it [Tang et al., 2016a,b]. These approaches mainly use attention-augmented RNNs for solving the task. However, they require the location of the aspect to be known in advance and therefore are only useful in pipeline models, while instead we model aspect extraction and sentiment classification as a joint task or using multi-tasking.

AESC has also often been tackled as a sequence labeling problem, mainly using Conditional Random Fields (CRFs) [Mitchell et al., 2013]. To model the problem in this fashion, collapsed or sentiment-bearing IOB labels [Zhang et al., 2015] are used. Pipeline models (i.e. task-independent model ensembles) have also been extensively studied by the same authors. Xu et al. [2014] performed *AESC* by modeling the linking relation between aspects and the sentiment-bearing phrases.

When it comes to the video review domain, there is related work on YouTube mining, mainly focused on exploiting user comments. For example, Wu et al. [2014] exploited crowdsourced textual data from time-synced commented videos, proposing a temporal topic model based on LDA. Tahara et al. [2010] introduced a similar approach for *Nico Nico*, using time-indexed social annotations to search for desirable scenes inside videos.

On the other hand, Severyn et al. [2014] proposed a systematic approach to mine user comments that relies on tree kernel models. Additionally, Krishna et al. [2013] performed sentiment analysis on YouTube comments related to popular topics using machine learning techniques, showing that the trends in users' sentiments is well correlated to the corresponding real-world events. Siersdorfer et al. [2010] presented an analysis of dependencies between comments and comment ratings, proving that community feedback in combination with term features in comments can be used for automatically determining the community acceptance of comments.

We also find some papers that have successfully attempted to use closed caption mining for video activity recognition [Gupta and Mooney, 2010] and scene segmentation [Gupta and Mooney, 2009]. Similar work has been done using closed captions to classify movies by genre [Brezeale and Cook, 2006] and summarize video programs [Brezeale and Cook, 2006]. Regarding multi-modal approaches for sentiment analysis, we see that previous work has focused mainly on sentiment classification, or the related task of emotion detection [Lakomkin et al., 2017], where the CMU MOSI dataset [Zadeh et al., 2016] appears as the main resource. In this setting, the main problem is how to model and capture cross-modality interactions to predict the sentiment correctly. In this regard Zadeh et al. [2017] proposed a tensor fusion layer that can better capture cross-modality interactions between text, audio and video inputs, while Poria et al. [2017] modeled inter-dependencies across difference utterances of a single video, obtaining further improvements.

Blanchard et al. [2018] are the first to tackle scalable multi-modal sentiment classification using both visual and acoustic modalities. More recently Ghosal et al. [2018] proposed an RNN-based multi-modal approach that relies on attention to learn the contributing features among multi-utterance representations. On the other hand Pham et al. [2018] introduced multi-modal sequence-to-sequence models which perform specially well in bi-modal settings. Akhtar et al. [2019] proposed a multi-modal, multi-task approach in which the inputs from a video (text, acoustic and visual frames), are exploited for simultaneously predicting the sentiment and expressed emotions of an utterance. Our work is related to all of these approaches, but it is different in that we apply multi-modal techniques not only for sentiment classification, but also for aspect extraction.

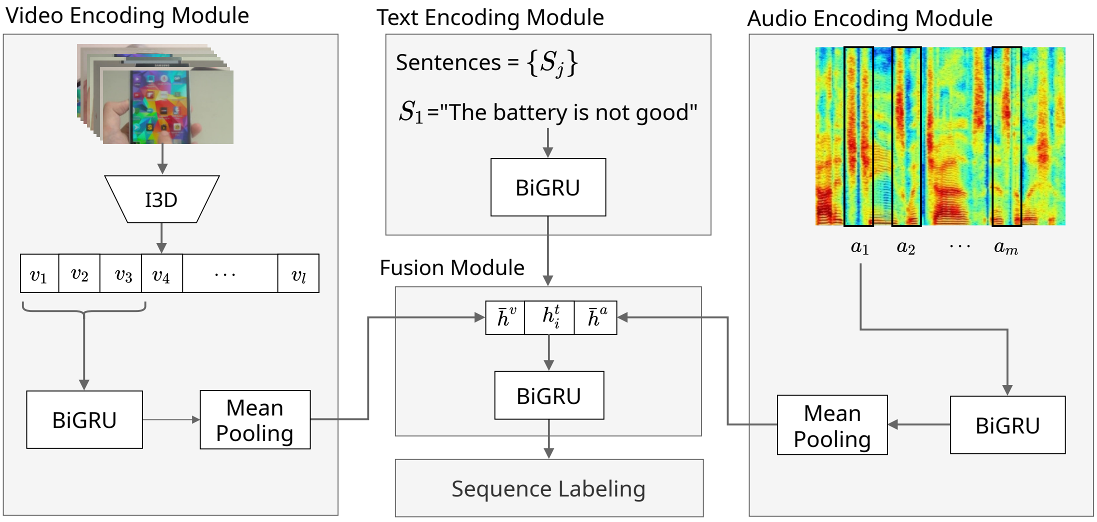


Figure 6.1: **Overview of our proposed approach for multi-modal opinion mining**. We use the three inherent modules to process the inherent information in a video. First the text encoding module that generates a reprentation of the transcript of the video. Then, the audio and video encoding modules that creates video representation for the audio and frames, respectively. Finally, we use a fusion module to fuse the modalities with the natural language and obtain the sequence labeling.

Finally, Marrese-Taylor et al. [2017] and Garcia et al. [2019b] contributed multi-modal datasets obtained from product and movie reviews respectively, specifically for the task of fine-grained opinion mining. Furthermore, Garcia et al. [2019a] recently used the latter to propose a hierarchical multi-modal model for opinion mining. Compared to them, our approach follows a more traditional setting for fine-grained opinion mining, while also offering a more general framework for the problem. Garcia et al. [2019a] utilize a single encoder that receives as input the concatenation of the features for each modality, for each token. This requires explicit alignment between the features of the different modalities at the token level. In contrast, since each modality is encoded separately in our approach, we only require the feature alignment to be at the sentence level.

6.3 Task Description

Opinion mining can be performed at several levels of granularity, the most common ones being the sentence level, and the more fine-grained aspect level. Fine-grained opinion mining can be further subdivided in two tasks: aspect extraction and aspect-level sentiment classification. The former deals with finding the aspects being referred to, and the latter with associating them with a sentiment.

Previous work usually casts this task as a sequence-labeling problem, where models have to predict whether a token is a part of an aspect and infer its sentiment polarity [Mitchell et al., 2013; Zhang et al., 2015; Liu et al., 2015]. Depending on the

	I	**love**	**the**	**saturated**	**colors**	**!**
$\mathbb{L}^{AE}$	O	O	O	B	I	O
$\mathbb{L}^{SC}$	ϕ	ϕ	ϕ	$+$	$+$	ϕ
$\mathbb{L}^{C}$	O	O	O	$B+$	$I+$	O

Table 6.1: Label definition alternatives for the tasks in ABSA using sequence labeling.

dataset annotations, aspect categories are in some cases specified as well.

Formally, given a sentence $s = [x_1, \dots, x_n]$, we want to automatically annotate each token x_i with its aspect membership and polarity. In the simpler case where we only want to perform Aspect Extraction, a common annotation scheme is to tag each token with a label $y_i \in \mathbb{L}^{AE}$ where $\mathbb{L}^{AE} = \{I, O, B\}$. In this scheme, commonly known as IOB, O labels indicate that a token is not a member of an aspect, B labels indicate that a token is at the beginning of an aspect, and I labels indicate that the token is inside an aspect.

Similarly, performing token-level Sentiment Classification only is equivalent to tagging each token with a label $y_i \in \mathbb{L}^{SC}$ where $\mathbb{L}^{SC} = \{\phi, +, -\}$, and ϕ denotes no sentiment, $+$ denotes a positive polarity and $-$ a negative one.

It is also possible to define a *collapsed* annotation scheme, where aspect membership and sentiment polarity are encoded in a single tag. We define the label set for this setting as $\mathbb{L}^{C} = \{O, B+, B-, I+, I-\}$.

Table 6.1 shows the possible ways to annotate the sentence "I love the saturated colors!" under these three annotation schemes, where the aspect being referred to is "saturated colors".

Labels can be further augmented with type information. For example Liu et al. [2015] used different tags for opinion targets (e.g. B-TARG), and opinion expressions (e.g., B-EXPR), however, we do not rely on this information.

6.4 Proposed Approach

We propose a multi-modal approach for aspect extraction and sentiment classification that leverages video, audio and textual features. This approach assumes we have a video review v containing opinions, its extracted audio stream a, and a transcription of the audio into a sequence of sentences $\mathbb{S}$. Further, each sentence $s \in \mathbb{S}$ is annotated with its respective start and end times in the video effectively mapping them to a video segment $v^s \subset v$ and its corresponding audio segment $a^s \subset a$. These segments do not necessarily cover the whole video, i.e., $\cup_{s=1}^{\mathbb{S}} v^s \subseteq v$ since the reviews may include parts that have no speech and therefore no sentences are associated to those. Our end goal is to produce a sequence of labels $l = [y_1, \dots, y_n]$ for each sentence $s = [x_1, \dots, x_n]$ while exploiting the information contained in v^s and a^s.

Figure 6.1 presents a high-level overview of our approach. We rely on an encoder-decoder paradigm to create separate representations for each modality [Cho et al., 2014]. The text encoding module generates a representation for each token in the input text, while the video and audio encoding layers produce utterance-level representations from each modality.

We propose combining these representations with an approach inspired by early-fusion [Xu et al., 2019], which allows for the word-level representations to interact with audio and visual features. Finally, a sequence labeling module is in charge of taking the final token-level representations and producing a token-level label. In the following sub-sections we describe each component of our model.

6.4.1 Text Encoding Module

This module generates a representation of the natural language input so that the obtained representation is useful for the sequence labeling task. Our text encoder first maps each word x_i into an embedded input sequence $\boldsymbol{x} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, then projects this into a vector $\boldsymbol{h}_i^t \in \mathbb{R}^{d_t}$, where d_t corresponds to the hidden dimension of the obtained text representation. Although our text encoding module is generic, in this paper we implement it as a bi-directional GRU [Cho et al., 2014], on top of pre-trained word embeddings, specifically GloVe [Pennington et al., 2014], as follows.

$$\boldsymbol{h}_i^t = \text{BiGRU}(\boldsymbol{x}_i, \boldsymbol{h}_{i-1}^t) \tag{6.1}$$

6.4.2 Audio Encoding Module

We assume the existence of a finite set of time-ordered audio features $\boldsymbol{a} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m]$ extracted from each audio utterance a^s, for instance with the procedure described in Section 6.5.2. We feed these vectors into another bi-directional GRU to add context to each time step, obtaining hidden states $\boldsymbol{h}_j^a \in \mathbb{R}^{d_a}$.

$$\boldsymbol{h}_j^a = \text{BiGRU}(\boldsymbol{a}_j, \boldsymbol{h}_{j-1}^a) \tag{6.2}$$

To obtain a condensed representation from the audio signal we again utilize mean pooling over the intermediate memory vectors, obtaining $\bar{\boldsymbol{h}}^a$.

6.4.3 Video Encoding Module

We propose a video encoding layer that generates a visual representation summarizing spatio-temporal patterns directly from the raw input frames. Concretely, given a video segment $\boldsymbol{v} = [v_1, \ldots, v_T]$, where $\boldsymbol{v}_i$ is a vector representing a single frame in v^s, our encoding module first maps this sequence into another sequence of video features $\hat{\boldsymbol{v}} = [\hat{v}_1, \ldots, \hat{v}_l]$ following the method described in Section 6.5.2. Later, this new sequence is mapped into a vector $\bar{\boldsymbol{h}}^v \in \mathbb{R}^{d_v}$ that captures summarized high-level visual semantics in the video, as follows:

$$\boldsymbol{h}_k^v = \text{BiGRU}(\hat{\boldsymbol{v}}_k, \boldsymbol{h}_{k-1}^v) \tag{6.3}$$

6.4.4 Fusion Module

We utilize an early fusion strategy similar to Xu et al. [2019] to aggregate the representations obtained from each modality. We concatenate the contextualized representation $\boldsymbol{h}_i^t$ for each token to the summarized representations of the additional modalities, $\bar{\boldsymbol{h}}^a$ and $\bar{\boldsymbol{h}}^v$, and feed this final vector representation to an additional BiGRU:

$$\boldsymbol{h}_i = \text{BiGRU}([\boldsymbol{h}_i^t; \bar{\boldsymbol{h}}^a; \bar{\boldsymbol{h}}^v], \boldsymbol{h}_{i-1}) \tag{6.4}$$

As a result, our model now allows the representation of each word in the input sentence to interact with the audio and visual features, enabling it to learn potentially different ways to associate each word with the additional modalities. An alternative way to achieve this would be to utilize attention mechanisms to enforce such association behavior, however, we instead let the model learn this relation without using any additional inductive bias.

6.4.5 Sequence Labeling Module

The main labeling module is a multi-layer perceptron guided by a self attention component. The self attention component enriches the representation $\boldsymbol{h}_i$ with contextual information coming from every other sequence element by performing the following operations:

$$u_{i,j} = \boldsymbol{v}_\alpha^\top \tanh(\boldsymbol{W}_\alpha[\boldsymbol{h}_i; \boldsymbol{h}_j] + \boldsymbol{b}_\alpha) \tag{6.5}$$

$$\alpha_{i,j} = \text{softmax}(u_{i,j}) \tag{6.6}$$

$$\boldsymbol{t}_i = \sum_{j=1}^{n} \alpha_{i,j} \cdot \boldsymbol{h}_j \tag{6.7}$$

$$\boldsymbol{o}_i = \boldsymbol{W}_l[\boldsymbol{h}_i; \boldsymbol{t}_i] + \boldsymbol{b}_l \tag{6.8}$$

where $\boldsymbol{o}_i$ is a vector associated to input x_i, and $\boldsymbol{v}_\alpha$, $\boldsymbol{W}_\alpha$, $\boldsymbol{W}_l$ and $\boldsymbol{b}_\alpha$, $\boldsymbol{b}_l$ are trainable parameters. As shown, these vectors are obtained using both the corresponding *aligned* input $\boldsymbol{h}_i$ and the attention-weighted vector $\boldsymbol{t}_i$.

Following previous work, we feed these vectors into a Linear Chain CRF layer, which performs the final labeling. Neural CRFs have proven to be especially effective for various sequence segmentation or labeling tasks in NLP [Ma and Hovy, 2016; Yang and Zhang, 2018; Yang et al., 2018], and have also been used successfully in the past for open domain opinion mining [Zhang et al., 2015]. Concretely, we model emission and transition potentials as follows,

$$\psi_i := e(x_i, y_i; \theta) = \boldsymbol{h}_i \cdot \boldsymbol{y}_i \tag{6.9}$$

$$\psi_{i,j} := q(y_i, y_j; \boldsymbol{\Pi}) = \boldsymbol{\Pi}_{y_i, y_j} \tag{6.10}$$

where $\boldsymbol{h}_i$ is the fused hidden state for position i and θ denotes the parameters involved in computing this vector, $\boldsymbol{y}_i$ is a one-hot vector associated to y_i, and $\boldsymbol{\Pi}$ is a trainable matrix of size $\mathbb{L}^{AE}$ or $\mathbb{L}^{C}$ depending on the setting —see Section 6.5 for more details on this. The score function of a given input sentence s and output sequence of labels l is defined as:

$$\Phi(s, l) = \sum_{i=1}^{n} \log e(x, y_i; \theta) + \log q(y_i, y_{i-1}; \mathbf{\Pi}) \tag{6.11}$$

In this work we directly optimize the negative log-likelihood associated to this score during training, and apply Viterbi decoding during inference to obtain the most likely labels.

6.5 Experimental Setup

We evaluate our proposal in several experimental settings based on previous work.

- **Simple**: We only focus on the task of aspect extraction, following a sequence labeling approach with regular IOB tags in $\mathbb{L}^{\text{AE}}$.
- **Collapsed Aspect-Level (CAL)**: We perform aspect extraction and aspect-level sentiment classification with a sequence labeling model, utilizing sentiment-bearing IOB tags in $\mathbb{L}^{\text{C}}$.
- **Collapsed Sentence-Level (CSL)**: Like the previous setting, but we only keep sentence examples that contain a single sentiment, so we can perform sentence-level sentiment classification. Again, we use sequence labeling with sentiment-bearing IOB tags in $\mathbb{L}^{\text{C}}$.
- **Joint Sentence-Level (JSL)**: We use a multi-tasking approach and perform sequence labeling for aspect extraction with regular IOB tags in $\mathbb{L}^{\text{AE}}$, and sequence classification to predict the sentence-level sentiment. In this sense, we add a final 3-layer fully-connected neural network that receives a mean-pooled representation of the fusion layer $\bar{\boldsymbol{h}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{h}_i$ and predicts a sentence-level sentiment. As loss function we utilize the mini-batch average cross-entropy with the gold standard class label. The total loss is the sum of the losses for sequence labeling and sequence classification.

Previous work has also shown that most sentences present a single aspect, and therefore a single sentiment [Marrese-Taylor et al., 2017; Zuo et al., 2018; Zhao et al., 2010], which motivates the introduction of the CSL and JSL settings. For these cases we filtered out sentences that do not fit this description.

6.5.1 Data

We report results on two different datasets containing fine-grained annotations for both opinion targets and sentiment.

First, we work with the Youtubean dataset [Marrese-Taylor et al., 2017], which contains sentences extracted from YouTube video annotated with aspects and their

```
168
00:20:41,150 --> 00:20:45,109
- How did he do that?
- Made him an offer he could not refuse.
```

Figure 6.2: Excerpt of a subtitle chunk (in SubRip format,) showing its main components.

respective sentiments. The data comes from the user-provided closed-captions derived from 7 different long product review videos about a cell phone, totaling up to 71 minutes of audiovisual data. In total there are 578 long sentences from free spoken descriptions of the product, on average each sentence consist of 20 words. The dataset has a total of 525 aspects, with more than 66% of the sentences containing at least one mention.

Second, we work with the fine-grained annotations gathered for the POM dataset by Garcia et al. [2019b]. This dataset is composed of 1000 videos containing reviews where a single speaker in frontal view makes a critique of a movie that he/she has watched. There are videos from 372 unique speakers, with 600 different movie titles being reviewed. Each video has an average length of about 94 seconds and contains 15.1 sentences on average. The fine-grained annotations we utilize are available for each token indicating if it is responsible for the understanding of the polarity of the sentence, and whether it describes the target of an opinion; each sentence has an average of 22.5 tokens. We assume that whenever there is an overlap between the span annotations for a given target and a certain polarity, the corresponding polarity can be assigned to that target, otherwise it is labeled as neutral.

Since the annotated sentences in both datasets are not associated to specific timestamps, in this work we propose a method based on heuristics to rescue the video segments that correspond to each annotated sentence by leveraging video subtitles (or closed-captions.)

As shown in Figure 6.2, closed captions or subtitles are composed of chunks that contain: (1) A numeric counter identifying each chunk, (2) The time at which the subtitle should appear on the screen followed by -> and the time when it should disappear, (3) The subtitle text itself on one or more lines, and (4) A blank line containing no text, indicating the end of this subtitle. These chunks exhibit a large variance in terms of their length, meaning that sentences are usually split into many chunks.

Starting from a subtitle file associated to a given product review video, we apply a fuzzy-matching approach between each annotated sentence for that review and each closed caption chunk. This is repeated for each one of the videos in our datasets. Whenever an annotated sentence matches exactly or has over 90% similarity with a closed caption chunk, its time-span is associated to that sentence. Finally, the "start" and "end" timestamps assigned to each sentence are defined by the start and end time spans of their first and last associated closed captions, sorted by time.

6.5.2 Implementation Details

Pre-processing for the natural language input is performed utilizing spacy[2], which we use mainly to tokenize. Input sentences are trimmed to a maximum length of 300 tokens, and tokens with frequency lower than 1 are replaced with a special *UNK* marker. To work with the POM dataset, which is already tokenized, we first convert it to the ABSA format, which is tokenization agnostic, and then we process it.

Although our audio encoder is generic, in this work we follow Lakomkin et al. [2017] and use Fast Fourier Transform spectrograms to extract rich vectors from each audio segment. Specifically, we use a window length of 1024 points and 512 points overlap, giving us vectors of size 513. Alternative audio feature extractors such as Degottex et al. [2014] could also be utilized.

On the other hand, we model video feature extraction using I3D [Carreira and Zisserman, 2017]. This method inflates the 2D filters of a well-known network e.g. Inception [Szegedy et al., 2015; Ioffe and Szegedy, 2015] or ResNet [He et al., 2016] for image classification to obtain 3D filters, helping us better exploit the spatio-temporal nature of video. We first pre-process the videos by extracting features of size 1024 using I3D with average pooling, taking as input the raw frames of dimension 256 × 256, at 25 fps. We use the model pre-trained on the kinetics400 dataset [Kay et al., 2017] released by the same authors. Despite our choice to obtain video features, again we note that our video encoder is generic, so other alternatives such as C3D [Tran et al., 2015] could be utilized.

Finally, all of our models are trained in an end-to-end fashion using Adam [Kingma and Ba, 2015] with a learning rate of 10^{-3}. To prevent over-fitting, we add dropout to the text encoding layer. We use a batch size of 8 for the Youtubean dataset, and of 64 for the POM dataset. The language encoder uses a hidden state of size 150, and we fine-tune the pre-trained GloVe.

On each case we compare the performance of our proposed approach against a baseline model that does not consider multi-modality, does not utilize pre-trained GloVe word embeddings and is based on a cross-entropy loss, in which case we simply utilize the mini-batch average cross-entropy between $\hat{\boldsymbol{y}}_i = \text{softmax}(\boldsymbol{o}_i)$ and the gold standard one-hot encoded labels $\boldsymbol{y}_i$, a vector that is the size of the tag label vocabulary for the corresponding task.

6.5.3 Evaluation

Since the size of Youtubean is relatively small, all our experiments in this dataset are evaluated using 5-fold cross validation. In the case of the POM dataset, we report performance on the validation and test sets averaging results for 5 different random seeds. In both cases we compare models using paired two-sided t-tests to check for statistical significance of the differences.

To evaluate our sequence labeling tasks we used the CoNLL *conlleval* script, taking the aspect extraction F1-score as our model selection metric for early stopping. To

[2] https://spacy.io

Setting	Model	Aspect Extraction			Sentiment Classification		
		P	**R**	**F1**	**P**	**R**	**F1**
Simple	Baseline	0.531	0.542	0.533	-	-	-
	Ours	**0.602****	**0.568**	**0.584*****	-	-	-
CAL	Baseline	0.546	0.538	0.539	0.710	0.688	0.696
	Ours	**0.590**	**0.572**	**0.581***	**0.722**	**0.722**	**0.718**
CSL	Baseline	0.526	0.463	0.490	**0.746**	**0.722**	**0.724**
	Ours	**0.563**	**0.581*****	**0.568****	0.720	0.674	0.688
JSL	Baseline	0.483	0.521	0.496	**0.946**	**0.946**	**0.946**
	Ours	**0.544*****	**0.552**	**0.545*****	**0.946**	**0.946**	**0.946**

Table 6.2: Summary of our results on the Youtubean dataset, *** denotes statistical significance at 99% confidence, ** at 95% and * at 90%.

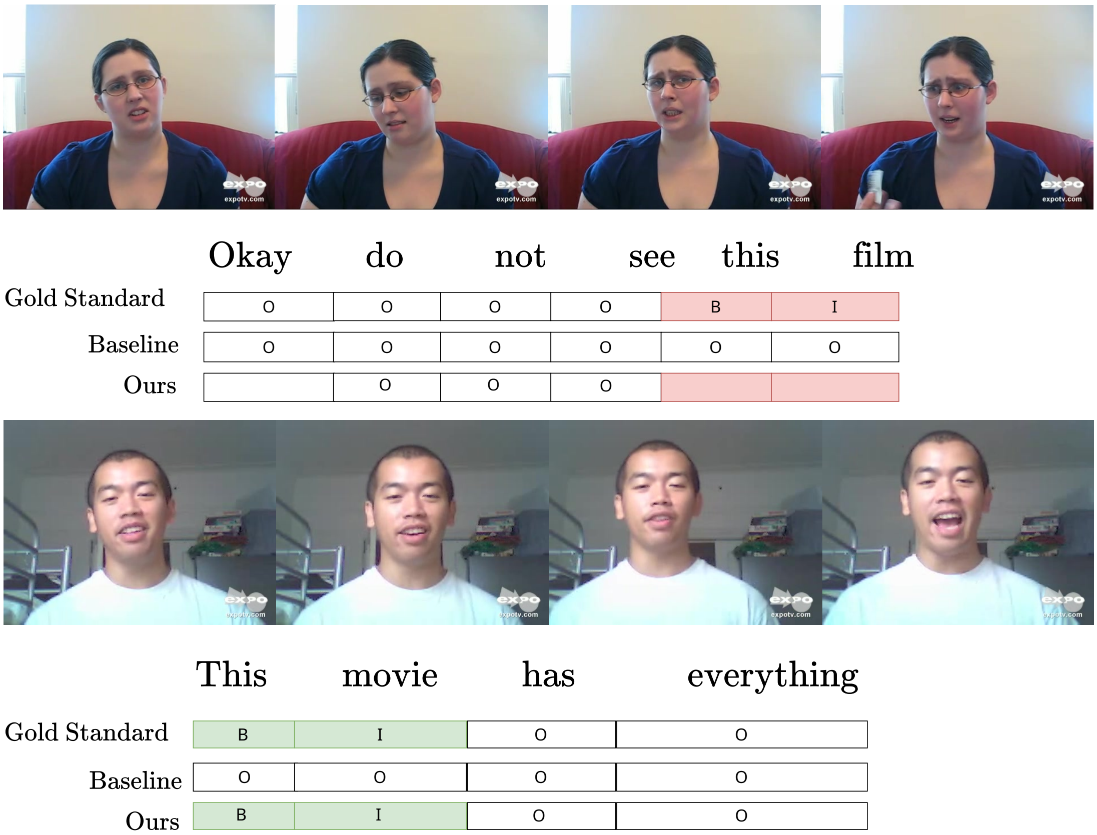
perform joint aspect extraction and sentiment classification, we considered *positive*, *negative* and *neutral* as sentiment classes, and decoupled the IOB collapsed tags using simple heuristics. Concretely, we recover the aspect extraction F1-score as well as classification performances for each sentiment class.

6.6 Results

To evaluate the effectiveness of our proposals, we perform several ablation studies on the *Simple* setting for the Youtubean dataset. Using variations of our baseline with pre-trained GLoVe embeddings (GV), conditional random field (CRF), audio and video modalities (A+V). Experiments are also performed using 5-fold cross-validation, and comparisons are always tested for significance using paired two-sided t-tests.

As Table 6.4 shows, although every proposed model variation performs better than the baseline, only the model uses video and audio modalities obtains a statistically superior performance. We also see that our proposed multi-modal variation is the one that obtains the best performance, also being statistically significant at the highest level of confidence. We believe these results show that our proposed multi-modal architecture is not only able to exploit the features in the audio and video inputs, but it can also leverage the information in the pre-trained word embeddings and benefit from having an inductive bias that is tailored for the task at hand, in this case, with a loss based on structured prediction for sequence labeling.

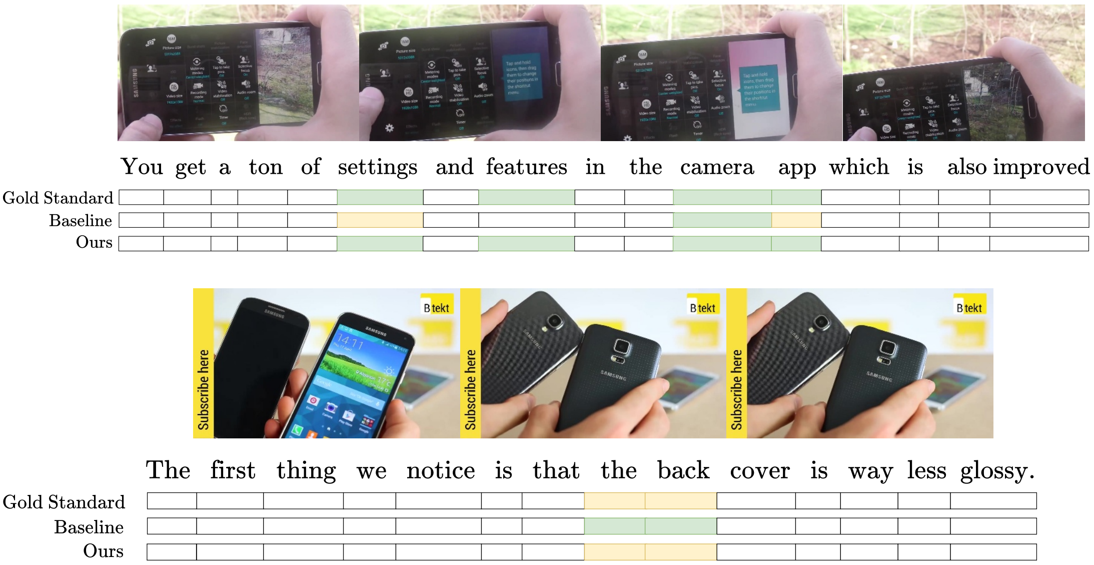
Table 6.2 summarizes our results for the Youtubean dataset, where we can see that our proposed multi-modal approach is able to outperform the baseline model for all settings in the aspect extraction task. When it comes to sentiment classification, our multi-modal approaches do not obtain significant performance gain in all cases, sometimes performing worse although without statistical significance. We also



	Okay	do	not	see	this	film
Gold Standard	O	O	O	O	🟥 B	🟥 I
Baseline	O	O	O	O	O	O
Ours		O	O	O	🟥	🟥

	This	movie	has	everything
Gold Standard	🟩 B	🟩 I	O	O
Baseline	O	O	O	O
Ours	🟩 B	🟩 I	O	O

Figure 6.3: Qualitative comparison between baseline and our method on the POM dataset. Green and red boxes represent positive and negative sentiment respectively.



	You	get	a	ton	of	settings	and	features	in	the	camera	app	which	is	also	improved
Gold Standard						🟩		🟩			🟩	🟩				
Baseline						🟨		🟩			🟩	🟨				
Ours						🟩		🟩			🟩	🟩				

	The	first	thing	we	notice	is	that	the	back	cover	is	way	less	glossy.
Gold Standard								🟨	🟨					
Baseline								🟩	🟩					
Ours								🟨	🟨					

Figure 6.4: Qualitative comparison between baseline and our method on the Youtubean dataset. Green and yellow boxes represent positive and neutral sentiment respectively.

Setting	**Model**	**Aspect Extraction**			**Sentiment Classification**		
		P	**R**	**F1**	**P**	**R**	**F1**
Simple	Baseline	0.394	0.379	0.386	-	-	-
	Ours	**0.396**	**0.406**	**0.399**	-	-	-
CAL	Baseline	0.364	**0.401***	0.382	**0.540*****	0.416	0.270
	Ours	**0.444****	0.368	**0.402****	0.488	**0.466*****	**0.342*****
CSL	Baseline	0.387	0.375	**0.408***	**0.614**	**0.446**	0.296
	Ours	**0.438***	**0.378**	0.404	0.532	**0.446**	**0.304**
JSL	Baseline	0.381	0.357	0.367	0.798	0.802	0.788
	Ours	**0.442*****	**0.401***	**0.420***	**0.924*****	**0.924*****	**0.922*****

Table 6.3: Summary of our results for the test set of the POM dataset, *** denotes statistical significance at 99% confidence, ** at 95% and * at 90%.

compare our results to the performance reported by Marrese-Taylor et al. [2017], who experimented on the *Simple* and *CSL* settings. Their models also use pre-trained word embedding —although different from GloVe— and as input they additionally receives binary features derived from POS tags and other word-level cues. We note, however, that they only experimented with a maximum length of 200 tokens, which makes our results not directly comparable. Their performance on aspect extraction for the *Simple* and *CAL* tasks are 0.561 and 0.555 F1-Score respectively, both of which are lower than ours. In terms of sentiment classification, they report results for each sentiment class with F1-Scores of 0.523, 0.149 and 0.811 for the positive, negative and neutral classes, respectively. Our model is able to outperform this baseline, with a cross-class average F1-Score of 0.718. We do not deepen the analysis in this regard, as numbers are difficult to interpret without statistical testing.

Tables 6.5 and 6.3 summarise our results for the *POM* dataset for the validation and test splits, respectively. Compared to the previous dataset we see similar results where our multi-modal approach consistently outperforms the baseline for aspect extraction, but with the gains being comparatively smaller. We also see that our model is able to significantly outperform the baseline in the sentiment classification tasks at least in two of out the three settings. In terms of previous work, our results cannot be directly compared to Garcia et al. [2019a] and Garcia et al. [2019b] as their problem setting is different from ours.

On a more broad perspective, we think the performance differences across datasets are related to the nature of each dataset. Meanwhile Youtubean contains reviews about actual physical products, which are often shown in the videos at the same time the reviewer is speaking, the POM dataset contains movie reviews where the speakers directly face the camera during most of the video, without utilizing any additional support material. As a result, the video reviews in the Youtubean dataset mainly focus on capturing images of the products under discussion, with relatively fewer scenes showing the reviewer. This means that there may be few visual cues

Model	Aspect Extraction		
	P	**R**	**F1**
T	0.532	0.543	0.533
T + CRF	0.558	0.528	0.541
T + GV	0.562	0.537	0.548
T + GV + CRF	0.576*	0.569	0.571**
T + A + V	0.587*	0.578	0.580*
T + CRF + A + V	0.578	0.570	0.573*
T + GV + CRF + A + V	0.602**	0.568	0.584***

Table 6.4: Ablation study on aspect extraction on the simple setting. *** denotes differences against the only text model (T) results are statistically significant at 99% confidence, ** at 95% and * at 90%. (A + V) refers to the audio and video modalities, (GV) stands for GLoVe embeddings and (CRF) for the model trained using the Conditional Random Fields loss.

Setting	Model	AE F1	SC F1
Simple	Baseline	0.428	-
	Ours	**0.433**	-
CAL	Baseline	0.412	0.240
	Ours	**0.427*****	**0.310****
CSL	Baseline	0.408	**0.264**
	Ours	**0.423***	0.262
JSL	Baseline	0.387	**0.950*****
	Ours	**0.469****	0.840

Table 6.5: Results for the validation set of the POM dataset, where *** denotes results are statistically significant at 99% confidence, ** at 95% and * at 90%.

in the manner of facial expressions or other specific actions that the models could exploit in order to perform better at the sentiment classification task, but more cues useful for aspect extraction. This situation is reverted in the POM dataset, which could explain why our models tend to perform better for sentiment classification, but offering smaller gains for the AE task.

We also think performance differences across datasets are to some extent explained by the nature of the annotations on each case. The annotation guidelines utilized to elaborate each dataset are actually quite different, with the annotations in the Youtubean dataset closely following those of the well-known SemEval datasets, which are target-centric and the POM standards substantially diverging from this. Concretely, Garcia et al. [2019b] propose a two-level annotation method, where "the smallest span of words that contains all the words necessary for the recognition of an opinion" are to be annotated. As a result, aspects annotated in the POM dataset often include pronouns which are more difficult to identify as aspects, often requiring co-reference resolution. With regards to aspect polarity, while it can be extracted directly from the Youtubean annotations, in the case of POM we needed some pre-processing as target and sentiment are annotated using independent text spans.

Qualitative results of the POM and Youtubean dataset in a multitask CAL can be seen in Figure 6.3 and 6.4 respectively, results suggest that the method learn to use the information from additional modalities and enhance the sentiment and aspect prediction.

Finally, as we observe that our models tend to obtain bigger gains on the AE tasks rather than on SC, we think this behaviour can be partially attributed to the inductive bias of our model, which makes it specially suitable for sequence segmentation tasks.

6.7 Summary

In this chapter we have presented a multi-modal approach for fine-grained opinion mining, introducing a modular architecture that utilizes features derived from the audio, video frames and language transcription of video reviews to perform aspect extraction and sentiment classification at the sentence level. To test our proposals we have taken two datasets built upon video review transcriptions containing fine-grained opinions, and introduced a technique that leverages the video subtitles to associate timestamps to each annotated sentence. Our results offer empirical evidence showing that the additional modalities contain useful information that can be exploited by our models to offer increased performance for both aspect extraction and sentiment classification, consistently outperforming text-only baselines.

Chapter 7

Conclusion and Future Directions

This thesis focuses on video analysis for understanding human actions and interactions. We studied different tasks where we have to anticipate, localise and summarise human actions. To conclude our work, in this chapter, we summarise the main contributions of this thesis and discuss some open problems and exciting directions for future research.

7.1 Summary

In order to contextualise our research with the scientific literature needed to tackle these challenging problems, we provide a brief background in Chapter 2. We started our technical contribution in Chapter 3 introducing a method for action anticipation by forecasting motion representations, concretely dynamic images. Our approach hallucinates future dynamic images by observing an early portion of the videos. The intuition is that the generation of motion representations are easier and more meaningful for the task of action anticipation than still images. We propose two loss functions that encourage an autoencoder to forecast useful motion representations, taking advantage of the way that dynamic images are created. We also encourage the autoencoder to produce discriminative motion representations tailored to the action anticipation task. Using this approach, we achieved a state of the art performance in the action anticipation task. In this chapter, we learned the importance of temporal information for understanding human actions in videos.

Then we studied the problem of temporal moment localisation using a natural-language query. In contrast with the previous task where we use only visual information, in this task, we use natural language to guide the localisation of action or moment in long untrimmed videos. In Chapter 4, we presented a proposal-free method. In contrast with the propose and ranking approaches, where methods create or use predefined clips as candidates, we introduce a proposal-free approach that localises the query by looking at the whole video at once. We also consider the subjectivity of the temporal annotations and propose a soft-labelling using a categorical distribution centred on the annotated start and end. Using this approach, we achieved a state of the art performance in the task.

Qualitative analysis in Chapter 4 suggests that when the method is confused it is because it does not consider any spatial information. In light of this, in Chapter 5, we introduce a new approach based on a spatial-temporal graph. The principal motivation of this work is to capture the human-human and human-objects relationships with the activity. We proposed a unique spatial graph that models such relationships conditioned in the input query. We create three semantically meaningful nodes for visual features, Human, Object, and Activity nodes. Human and Object nodes receive the observations made by an object detector mechanism, and the Activity node receives an activity representation using a 3D convolutional neural network.

Moreover, we create three different language nodes that model the relationship at the language level of the "subject-object," "subject-verb," and "verb-object." We use a language-conditional message passing to send messages between nodes and create an improved representation of the activity used by a temporal graph to determine the start and end of the query. Using this approach, we achieved a state of the art performance in the temporal moment localisation task in different benchmarks.

In Chapter 6, the final technical chapter of this thesis, we studied the problem of fine-grained opinion mining in video review using a multi-modal setting. People use video reviews as a guide to answering what, why, and where to buy something. We tackled this problem using the three different modalities naturally present in a video —audio, appearance (frames), and language (speech transcript)— to determine the most relevant aspect of the object under review and the sentiment polarity of the reviewer upon that aspect. We proposed an early fusion mechanism of the three modalities. This approach allows to fuse the modalities at the sentence level, and it is a general framework that does not lay in any strict constraints on the individual encodings. Using this approach, we prove the benefit that vision and audio can bring to a task that has been typically tackled by the natural language community using only text.

7.2 Future Work

Our work is motivated by the vast applicability that video analysis and human action understanding can bring to improve our daily life, e.g., domestic robots will need action anticipations capabilities to help humans to perform a specific task. Search engines that understand the content of the video can help people to find where they leave objects in a house or localise important events in a video such as nocturnal seizures. Summarising the video content automatically to create useful visualisations that can help in the decision-making process of buying, creating or improving products. In this section, we present a discussion of potential future research directions towards understanding human action in videos to release all the potential in its applications.

7.2.1 Action anticipation: Uncertainty and Robotics

In Chapter 3, we described an action anticipation mechanism that forecasts motion representation recursively. Although the method reaches state of the art performance by forecasting dynamic images, there is still room to improve. One of the main problems of generating sequences of motion representations is the propagation of error through consecutive generations. Therefore, determining the number of dynamic images that maximise the action anticipation capabilities of our method is an exciting direction to explore. It is not only necessary for the method mentioned above. It could help many other video techniques that rely on forecast video frames or features. In our specific case, the use of discriminative motion representation allows us to use information gain to determine such numbers.

Although action anticipation in videos has received considerable attention in the last years, there are few empirical studies of using these approaches in real scenarios. Action anticipation for reactive robot response is among those [Koppula and Saxena, 2015]. It consists of predicting the human's action and executing the action of an agent accordingly. It could be used to prevent accidents [Aliakbarian et al., 2018] or to perform a task cooperatively [Villani et al., 2018]. However, these works focus on predicting an action without context or goal, i.e., an agent opening a refrigerator, since it perceives that a person is close to it, but they are not involved in a task together. We imagine a synthetic or real environment where the robot performs a task collaboratively with a human, such that each action that the robot undertakes is geared towards accomplishing a task, e.g., assemble furniture. In such a scenario, the robot's errors in anticipating the action could be catastrophic or just another path to accomplish the goal. This scenario can be considered as a partially observable Markov decision process, in which the human's future actions depend on the agent's actions. In this context, we believe that action anticipation is an essential tool to create a complete agent, an agent that can *see*, *communicate* and *act* [Anderson, 2018]. In our view, the current state of the art for the task of action anticipation is in the necessary stage to move to this next step, as well as the other subfields needed to create a complete agent. It is for that reason that we believe there is great potential for future researchers that might focus on creating complete agents that help people to accomplish a task.

7.2.2 Moment Localisation: Closing the Loop and Compositional Videos

In the case of temporal moment localisation, Chapters 4 and 5, we proposed a proposal-free method that can localise a query by seeing the whole video at once. The possibility to see the whole video allows us to think in methods that can close the loop between vision and language. For each query or description, there exists a temporal section in the video that is performing such a description. As also, for each moment in the video there is a natural language sentence that can describe the moment. This scenario is ideal for using cycle consistency approaches [Zhu et al., 2017; Felix et al., 2018], where one method localises a segment in the video that is related to the query and through another method, e.g., dense video captioning [Krishna et al., 2017], we generate the query for that segment. We believe that this direction can help both tasks, temporal moment localisation and dense video captioning, to generate more accurate localisations and diverse captions. However, it is likely that these types of methods will need a more sophisticated attention mechanism in the temporal domain.

Although the guided attention that we introduced in Chapter 4 is satisfying the needs to localise the query, it will possibly not work for caption generation. Softmax usually attends one or two features in the segment, which limitate the information that is transferred to a captioner. Thus, we believe the use of a structured attention mechanism that provides much broader information to the captioner and also helps the localisation [Kim et al., 2017; Qiu et al., 2020].

In the context of closing the loop, we can extend our spatial-temporal graph to solve the task of dependency parsing. Dependency parsing is the task of analysing the grammatical structure of a sentence and establishing the relationship between words [Nivre et al., 2016, 2007; Ji et al., 2019]. Our method uses a multi-head attention mechanism to attend to the words that capture the relationship between the subject, nouns and verbs. However, we have seen that the attention mechanism is not fully capturing such language relationships. The network uses what it thinks is the most valuable language relationships to localise the task. We believe that guiding the language module to feed the spatial-temporal graph with the correct relationship of the words could help the method to understand the video better. Moreover, we believe that an additional source of information that can help a dependency parsing method is the relationship between the activity, human and object visual nodes. This symbiotic relationship between visual and language should be exploited, and it is an exciting direction for future research.

Another technical future line of work is modelling the relationship between start and end. Currently, our proposed methods do not impose any constraint on the temporal boundaries. For example, one can impose the simple constraint that the start must be before the end. Others can capture the prior information of the duration of an action and impose this constraint in the start and end, taking in consideration the different speeds in a video.

Temporal moment localisation can be applied in many different tasks, which can help humans in their daily life. We believe one of the ideal scenarios where this method can be used is compositional videos. We imagine a method that generates new videos through the combination of different moments. For example, we create a new recipe which does not have a video associated with it. Ideally, we would use different moments that can compose the video recipe and create a completely new tutorial without the need of human intervention.

7.2.3 Video Opinion Mining: Gestures and Sarcasm

Our work in Chapter 6 demonstrated that the use of visual and audio features provide useful information for the opinion mining task. Our method is straightforward, and we believe it is the starting point in this direction of research. We believe that for better capturing the sentiment polarity of an opinion, we need to add fine-grained representations of the spatial-visual information in the videos, such as *faces* and *gestures*. In the same direction, adding a fine-grained representation of the *objects* could help in the aspect extraction, and we need a method that captures the relationship between objects and gestures in the scene.

In terms of the fusion mechanism, we think that aligning features from the different modalities to cope with their different information speed would be beneficial. Currently, we are concatenating the video and audio representation naively. We concatenate a rough representation of these modalities for each of the words in a transcript. Thus, a more sophisticated fusion mechanism that aligns video, audio and words in the transcript could better capture the aspect and sentiments in the speech.

We also think there is a need for more and better datasets for this task with more unconstrained and natural video reviews considering a more general setting with different types of objects.

7.3 Conclusion

This thesis has explored the challenge of video analysis for understanding human actions and interactions. Towards this end, we have leveraged the temporal and spatial information to propose methods that better capture the necessary information in a video to localise, forecast and summarise human actions. Nevertheless, as we discussed in previous sections, there is still much work to be done on improving these methods and there are many exciting directions in which they can be extended. We hope that our work can provide the direction on which further research can stand.

Bibliography

2015. Mexaction2 dataset. (2015). http://mexculture.cnam.fr/xwiki/bin/view/Datasets/Mex+action+dataset. (cited on page 42)

Ahad, M. A. R.; Tan, J. K.; Kim, H.; and Ishikawa, S., 2012. Motion history image: its variants and applications. *Machine Vision and Applications*, 23, 2 (2012), 255–281. (cited on page 24)

Akhtar, M. S.; Chauhan, D.; Ghosal, D.; Poria, S.; Ekbal, A.; and Bhattacharyya, P., 2019. Multi-task Learning for Multi-modal Emotion Recognition and Sentiment Analysis. In *Proceedings of the 2019 Conference of the North*, 370–379. Association for Computational Linguistics, Minneapolis, Minnesota. doi:10.18653/v1/N19-1034. http://aclweb.org/anthology/N19-1034. (cited on page 90)

Aliakbarian, M. S.; Saleh, F. S.; Salzmann, M.; Fernando, B.; Petersson, L.; and Andersson, L., 2018. Viena2: A driving anticipation dataset. In *Asian Conference on Computer Vision*, 449–466. Springer. (cited on page 105)

Aliakbarian, S.; Sadat Saleh, F.; Salzmann, M.; Fernando, B.; Petersson, L.; and Andersson, L., 2017. Encouraging lstms to anticipate actions very early. *ICCV*, (2017). (cited on pages 3, 24, 26, 27, 32, and 35)

Alwassel, H.; Caba Heilbron, F.; Escorcia, V.; and Ghanem, B., 2018. Diagnosing error in temporal action detectors. In *The European Conference on Computer Vision (ECCV)*. (cited on pages 41 and 47)

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L., 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*. (cited on page 65)

Anderson, P. J., 2018. *Vision and Language Learning: From Image Captioning and Visual Question Answering towards Embodied Agents*. Ph.D. thesis, College of Engineering and Computer Science, The Australian National University. (cited on page 105)

Baker, S. and Matthews, I., 2004. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56, 3 (2004), 221–255. (cited on page 12)

Baldi, P., 2012. Autoencoders, unsupervised learning, and deep architectures. In *ICML*. (cited on page 29)

Beinborn, L.; Botschen, T.; and Gurevych, I. Multimodal Grounding for Language Processing. 15. (cited on page 88)

Ben-Shabat, Y.; Yu, X.; Saleh, F. S.; Campbell, D.; Rodriguez-Opazo, C.; Li, H.; and Gould, S., 2021. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, (2021). (cited on page 9)

Bilen, H.; Fernando, B.; Gavves, E.; and Vedaldi, A., 2017. Action recognition with dynamic image networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, 99 (2017), 1–1. doi:10.1109/TPAMI.2017.2769085. (cited on pages 12, 24, 25, 27, 30, and 32)

Bilen, H.; Fernando, B.; Gavves, E.; Vedaldi, A.; and Gould, S., 2016. Dynamic image networks for action recognition. In *CVPR*. (cited on pages 12, 14, 23, 27, 30, 32, and 34)

Bishop, C. M. et al., 1995. *Neural networks for pattern recognition*. Oxford university press. (cited on page 1)

Blanchard, N.; Moreira, D.; Bharati, A.; and Scheirer, W., 2018. Getting the subtext without the text: Scalable multimodal sentiment classification from visual and acoustic modalities. In *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*, 1–10. Association for Computational Linguistics, Melbourne, Australia. doi:10.18653/v1/W18-3301. http://aclweb.org/anthology/W18-3301. (cited on page 90)

Blank, M.; Gorelick, L.; Shechtman, E.; Irani, M.; and Basri, R., 2005. Actions as space-time shapes. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, 1395–1402. IEEE. (cited on page 11)

Brezeale, D. and Cook, D., 2006. Using closed captions and visual features to classify movies by genre. In *Proceedings of the 7th International Workshop on Multimedia Data Mining (MDM/KDD06): Poster Session*. ACM, Washington, DC, USA. (cited on page 90)

Brox, T.; Bruhn, A.; Papenberg, N.; and Weickert, J., 2004. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, 25–36. Springer. (cited on page 12)

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*, 961–970. (cited on pages 21, 48, 60, and 68)

Carreira, J. and Zisserman, A., 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*. (cited on pages 12, 17, 25, 45, 49, 65, 69, and 97)

Chao, Y.; Vijayanarasimhan, S.; Seybold, B.; Ross, D. A.; Deng, J.; and Sukthankar, R., 2018. Rethinking the faster R-CNN architecture for temporal action localization. *CVPR*, (2018). (cited on pages 4 and 40)

Chen, D.; Fisch, A.; Weston, J.; and Bordes, A., 2017. Reading Wikipedia to Answer Open-Domain Questions. 1870–1879. doi:10.18653/v1/P17-1171. https://www.aclweb.org/anthology/papers/P/P17/P17-1171/. (cited on pages 41 and 88)

Chen, J.; Chen, X.; Ma, L.; Jie, Z.; and Chua, T.-S., 2018. Temporally grounding natural sentence in video. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 162–171. Association for Computational Linguistics, Brussels, Belgium. https://www.aclweb.org/anthology/D18-1015. (cited on pages 43, 51, 52, and 61)

Chen, S. and Jiang, Y.-G., 2019. Semantic proposal for activity localizaiton in videos via sentence query. *AAAI*, (2019). (cited on pages xix, 43, 51, 61, 72, and 73)

Chen, X.; Ritter, A.; Gupta, A.; and Mitchell, T., 2015. Sense Discovery via Co-Clustering on Images and Text. 5298–5306. https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Chen_Sense_Discovery_via_2015_CVPR_paper.html. (cited on page 88)

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y., 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. Association for Computational Linguistics, Doha, Qatar. http://www.aclweb.org/anthology/D14-1179. (cited on pages 20, 67, 72, 92, and 93)

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, (2014). (cited on pages 20, 45, 47, and 64)

Degottex, G.; Kane, J.; Drugman, T.; Raitio, T.; and Scherer, S., 2014. Covarep—a collaborative voice analysis repository for speech technologies. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, 960–964. IEEE. (cited on page 97)

Elliott, D.; Frank, S.; Barrault, L.; Bougares, F.; and Specia, L., 2017. Findings of the Second Shared Task on Multimodal Machine Translation and Multilingual Image Description. In *Proceedings of the Second Conference on Machine Translation*, 215–233. Association for Computational Linguistics, Copenhagen, Denmark. doi:10.18653/v1/W17-4718. http://aclweb.org/anthology/W17-4718. (cited on page 88)

Escorcia, V.; Caba Heilbron, F.; Niebles, J. C.; and Ghanem, B., 2016a. DAPs: Deep Action Proposals for Action Understanding. In *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, 768–784. Springer International Publishing. (cited on pages 4 and 42)

Escorcia, V.; Heilbron, F. C.; Niebles, J. C.; and Ghanem, B., 2016b. DAPs: Deep Action Proposals for Action Understanding.

ECCV, (2016). https://ivul.kaust.edu.sa/Documents/Publications/2016/DAPsDeepActionProposalsforActionUnderstanding.pdf. (cited on page 40)

Felix, R.; Kumar, V. B.; Reid, I.; and Carneiro, G., 2018. Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 21–37. (cited on page 105)

Fernando, B.; Gavves, E.; Oramas, J.; Ghodrati, A.; and Tuytelaars, T., 2017. Rank pooling for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 4 (2017), 773–787. (cited on pages 12 and 13)

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R., 2017a. Tall: Temporal activity localization via language query. In *ICCV*. (cited on pages 4, 40, 41, 42, 43, 45, 48, 51, 52, 60, 61, 68, 69, and 72)

Gao, J.; Yang, Z.; and Nevatia, R., 2017b. Red: Reinforced encoder-decoder networks for action anticipation. *arXiv:1707.04818*, (2017). (cited on page 27)

Gao, J.; Yang, Z.; Sun, C.; Chen, K.; and Nevatia, R., 2017c. TURN TAP: temporal unit regression network for temporal action proposals. *ICCV*, (2017). http://arxiv.org/abs/1703.06189. (cited on pages 4 and 40)

Garcia, A.; Colombo, P.; d'Alché-Buc, F.; Essid, S.; and Clavel, C., 2019a. From the Token to the Review: A Hierarchical Multimodal approach to Opinion Mining. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5542–5551. Association for Computational Linguistics, Hong Kong, China. doi:10.18653/v1/D19-1556. (cited on pages 91 and 100)

Garcia, A.; Essid, S.; d'Alché Buc, F.; and Clavel, C., 2019b. A multimodal movie review corpus for fine-grained opinion mining. *ArXiv*, (Feb. 2019). http://arxiv.org/abs/1902.10102. ArXiv: 1902.10102. (cited on pages 6, 21, 89, 91, 96, 100, and 101)

Gavrilyuk, K.; Ghodrati, A.; Li, Z.; and Snoek, C. G. M., 2018. Actor and action video segmentation from a sentence. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 46)

Ge, R.; Gao, J.; Chen, K.; and Nevatia, R., 2019. Mac: Mining activity concepts for language-based temporal localization. In *WACV*. (cited on pages 43, 45, 51, and 61)

Gerhardt, C.; Frobenius, M.; and Ley, S., 2013. *Culinary Linguistics*. John Benjamins Publishing. (cited on page 69)

Ghosal, D.; Akhtar, M. S.; Chauhan, D.; Poria, S.; Ekbal, A.; and Bhattacharyya, P., 2018. Contextual Inter-modal Attention for Multi-modal Sentiment Analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural*

Language Processing, 3454–3466. Association for Computational Linguistics, Brussels, Belgium. doi:10.18653/v1/D18-1382. http://aclweb.org/anthology/D18-1382. (cited on page 90)

Ghosh, S.; Agarwal, A.; Parekh, Z.; and Hauptmann, A. G., 2019. Excl: Extractive clip localization using natural language descriptions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1984–1990. (cited on pages 41, 43, 50, 51, 52, 60, 61, and 73)

Girdhar, R.; Carreira, J.; Doersch, C.; and Zisserman, A., 2019. Video Action Transformer Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 244–253. (cited on page 61)

Goodfellow, I.; Bengio, Y.; and Courville, A., 2016. *Deep learning*. MIT press. (cited on page 15)

Gu, C.; Sun, C.; Ross, D. A.; Vondrick, C.; Pantofaru, C.; Li, Y.; Vijayanarasimhan, S.; Toderici, G.; Ricco, S.; Sukthankar, R.; Schmid, C.; and Malik, J., 2018. AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6047–6056. (cited on page 61)

Gupta, S. and Mooney, R., 2009. Using closed captions to train activity recognizers that improve video retrieval. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, 30–37. doi:10.1109/CVPRW.2009.5204202. (cited on page 90)

Gupta, S. and Mooney, R. J., 2010. Using closed captions as supervision for video activity recognition. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2010)*, 1083–1088. Atlanta, GA. http://www.cs.utexas.edu/users/ai-lab/?gupta:aaai10. (cited on page 90)

Hahn, M.; Kadav, A.; Rehg, J. M.; and Graf, H. P., 2019. Tripping through time: Efficient localization of activities in videos. *arXiv preprint arXiv:1904.09936*, (2019). (cited on pages 43, 51, 52, and 72)

Hasan, M. K.; Rahman, W.; Bagher Zadeh, A.; Zhong, J.; Tanveer, M. I.; Morency, L.-P.; and Hoque, M. E., 2019. UR-FUNNY: A multimodal language dataset for understanding humor. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2046–2056. Association for Computational Linguistics, Hong Kong, China. doi:10.18653/v1/D19-1211. https://www.aclweb.org/anthology/D19-1211. (cited on pages 6 and 89)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2016. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 45, 65, and 97)

Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B., 2017. Localizing moments in video with natural language. In *ICCV*. (cited on pages 4, 40, 42, 45, 49, 51, 52, and 61)

Hendricks, L. A.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B., 2018. Localizing moments in video with temporal language. In *EMNLP*. (cited on page 43)

Hinton, G.; Vinyals, O.; and Dean, J., 2015. Distilling the knowledge in a neural network. *arXiv:1503.02531*, (2015). (cited on page 30)

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, 9, 8 (1997), 1735–1780. (cited on page 19)

Hong, Y.; Rodriguez-Opazo, C.; Qi, Y.; Wu, Q.; and Gould, S., 2020. Language and visual entity relationship graphfor agent navigation. (cited on page 9)

Hong*, Y.; Rodriguez-Opazo*, C.; Wu, Q.; and Gould, S., 2020. Sub-instruction aware vision-and-language navigation. (cited on page 9)

Hu, R.; Rohrbach, A.; Darrell, T.; and Saenko, K., 2019. Language-Conditioned Graph Networks for Relational Reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, 10294–10303. (cited on pages 60 and 62)

Idrees, H.; Zamir, A. R.; Jiang, Y.-G.; Gorban, A.; Laptev, I.; Sukthankar, R.; and Shah, M., 2017. The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155 (Feb. 2017), 1–23. doi:10.1016/j.cviu.2016.10.018. http://www.sciencedirect.com/science/article/pii/S1077314216301710. (cited on page 42)

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, (2015). (cited on pages 45, 65, and 97)

Irsoy, O. and Cardie, C., 2014. Opinion Mining with Deep Recurrent Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 720–728. Association for Computational Linguistics, Doha, Qatar. http://www.aclweb.org/anthology/D14-1080. (cited on pages 88 and 89)

Jain, A.; Singh, A.; Koppula, H. S.; Soh, S.; and Saxena, A., 2016a. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *ICRA*. (cited on pages 26 and 35)

Jain, A.; Zamir, A. R.; Savarese, S.; and Saxena, A., 2016b. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5308–5317. (cited on pages 62 and 64)

Jhuang, H.; Gall, J.; Zuffi, S.; Schmid, C.; and Black, M. J., 2013. Towards understanding action recognition. In *ICCV*. (cited on pages 21 and 32)

Ji, T.; Wu, Y.; and Lan, M., 2019. Graph-based dependency parsing with graph neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2475–2485. Association for Computational Linguistics, Florence, Italy. doi:10.18653/v1/P19-1237. https://www.aclweb.org/anthology/P19-1237. (cited on page 106)

Jia, X.; De Brabandere, B.; Tuytelaars, T.; and Gool, L. V., 2016. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, 667–675. (cited on page 46)

Jiang, B.; Huang, X.; Yang, C.; and Yuan, J., 2019. SLTFNet: A spatial and language-temporal tensor fusion network for video moment retrieval. *Information Processing & Management*, 56, 6 (Nov. 2019), 102104. doi:10.1016/j.ipm.2019.102104. (cited on page 62)

Jiang, Y.-G.; Bhattacharya, S.; Chang, S.-F.; and Shah, M., 2013. High-level event recognition in unconstrained videos. *International journal of multimedia information retrieval*, 2, 2 (2013), 73–101. (cited on page 60)

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 1725–1732. (cited on page 17)

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A., 2017. The kinetics human action video dataset. *CoRR*, (2017). http://arxiv.org/abs/1705.06950. (cited on pages 49 and 97)

Kim, Y.; Denton, C.; Hoang, L.; and Rush, A. M., 2017. Structured attention networks. *ICLR*, (2017). (cited on page 106)

Kingma, D. P. and Ba, J., 2015. Adam: A method for stochastic optimization. *ICLR*, (2015). (cited on pages 33, 49, 69, and 97)

Kingma, D. P.; Mohamed, S.; Rezende, D. J.; and Welling, M., 2014. Semi-supervised learning with deep generative models. In *NIPS*. (cited on page 29)

Kitani, K. M.; Ziebart, B. D.; Bagnell, J. A.; and Hebert, M., 2012. Activity forecasting. In *ECCV*. (cited on page 27)

Klaser, A., 2010. *Learning human actions in video*. Theses, Institut National Polytechnique de Grenoble - INPG. (cited on page 21)

Kong, Y.; Kit, D.; and Fu, Y., 2014. A discriminative model with multiple temporal scales for action prediction. In *ECCV*. (cited on page 26)

KOPPULA, H. S. AND SAXENA, A., 2015. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38, 1 (2015), 14–29. (cited on page 105)

KRISHNA, A.; ZAMBRENO, J.; AND KRISHNAN, S., 2013. Polarity trend analysis of public sentiment on youtube. In *Proceedings of the 19th International Conference on Management of Data*, COMAD '13 (Ahmedabad, India, 2013), 125–128. Computer Society of India, Mumbai, India, India. http://dl.acm.org/citation.cfm?id=2694476.2694505. (cited on page 90)

KRISHNA, R.; HATA, K.; REN, F.; FEI-FEI, L.; AND NIEBLES, J. C., 2017. Dense-captioning events in videos. In *ICCV*. (cited on pages 41, 48, 60, 68, and 105)

KRISHNA, R.; ZHU, Y.; GROTH, O.; JOHNSON, J.; HATA, K.; KRAVITZ, J.; CHEN, S.; KALANTIDIS, Y.; LI, L.-J.; SHAMMA, D. A.; BERNSTEIN, M.; AND FEI-FEI, L., 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations. https://arxiv.org/abs/1602.07332. (cited on pages 17 and 65)

KSCHISCHANG, F. R.; FREY, B. J.; AND LOELIGER, H.-A., 2001. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47, 2 (2001), 498–519. (cited on page 64)

KUEHNE, H.; JHUANG, H.; GARROTE, E.; POGGIO, T.; AND SERRE, T., 2011. Hmdb: a large video database for human motion recognition. In *ICCV*. (cited on page 32)

LAKOMKIN, E.; WEBER, C.; AND WERMTER, S., 2017. Automatically augmenting an emotion dataset improves classification using audio. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 194–197. Association for Computational Linguistics, Valencia, Spain. http://www.aclweb.org/anthology/E17-2031. (cited on pages 90 and 97)

LAVIERS, K.; SUKTHANKAR, G.; AHA, D. W.; MOLINEAUX, M.; DARKEN, C.; ET AL., 2009. Improving offensive performance through opponent modeling. In *AIIDE*. (cited on page 35)

LAZARIDOU, A.; PHAM, N. T.; AND BARONI, M., 2015. Combining Language and Vision with a Multimodal Skip-gram Model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 153–163. Association for Computational Linguistics, Denver, Colorado. doi:10.3115/v1/N15-1016. http://aclweb.org/anthology/N15-1016. (cited on page 88)

LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; AND JACKEL, L. D., 1989. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1, 4 (Dec. 1989), 541–551. doi:10.1162/neco.1989.1.4.541. https://doi.org/10.1162/neco.1989.1.4.541. (cited on pages 1 and 15)

Li, D.; Rodriguez, C.; Yu, X.; and Li, H., 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*. (cited on page 9)

Li, K. and Fu, Y., 2014. Prediction of human activity by discovering temporal sequence patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 8 (2014), 1644–1657. (cited on page 26)

Li, Z.; Tao, R.; Gavves, E.; Snoek, C. G.; and Smeulders, A. W., 2017. Tracking by natural language specification. In *CVPR*, 6495–6503. (cited on page 46)

Lin, J.; Mellish, C.; and Reiter, E. Style Variation in Cooking Recipes. 5. (cited on page 69)

Lin, T.; Zhao, X.; and Shou, Z., 2017. Single Shot Temporal Action Detection. In *Proceedings of the 25th ACM International Conference on Multimedia*, MM '17, 988–996. ACM, New York, NY, USA. doi:10.1145/3123266.3123343. http://doi.acm.org/10.1145/3123266.3123343. Event-place: Mountain View, California, USA. (cited on pages 4, 40, and 42)

Liu, M.; Wang, X.; Nie, L.; He, X.; Chen, B.; and Chua, T.-S., 2018. Attentive moment retrieval in videos. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 15–24. ACM. (cited on pages 43, 45, 51, and 61)

Liu, P.; Joty, S.; and Meng, H., 2015. Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1433–1443. Association for Computational Linguistics, Lisbon, Portugal. http://aclweb.org/anthology/D15-1168. (cited on pages 88, 89, 91, and 92)

Liu, Y.; Gupta, A.; Abbeel, P.; and Levine, S., 2019. Imitation from observation: Learning to imitate behaviors from raw video via context translation. (2019). (cited on pages 4 and 40)

Ma, S.; Sigal, L.; and Sclaroff, S., 2016a. Learning activity progression in lstms for activity detection and early detection. In *CVPR*. (cited on pages 3, 24, 26, 35, and 36)

Ma, S.; Sigal, L.; and Sclaroff, S., 2016b. Learning Activity Progression in LSTMs for Activity Detection and Early Detection. 1942–1950. http://openaccess.thecvf.com/content_cvpr_2016/html/Ma_Learning_Activity_Progression_CVPR_2016_paper.html. (cited on page 42)

Ma, X. and Hovy, E., 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1064–1074. Association for Computational Linguistics, Berlin, Germany. doi:10.18653/v1/P16-1101. (cited on page 94)

Marrese-Taylor, E.; Balazs, J.; and Matsuo, Y., 2017. Mining fine-grained opinions on closed captions of YouTube videos with an attention-RNN. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 102–111. Association for Computational Linguistics, Copenhagen, Denmark. http://www.aclweb.org/anthology/W17-5213. (cited on pages 6, 21, 88, 89, 91, 95, and 100)

Marrese-Taylor*, E.; Rodriguez-Opazo*, C.; Balazs, J.; Gould, S.; and Matsuo, Y., 2020. A multi-modal approach to fine-grained opinion mining on video reviews. In *Second Grand-Challenge and Workshop on Multimodal Language (Challenge-HML)*, 8–18. Association for Computational Linguistics, Seattle, USA. https://www.aclweb.org/anthology/2020.challengehml-1.2. (cited on page 9)

Mesnil, G.; He, X.; Deng, L.; and Bengio, Y., 2013. Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In *INTERSPEECH*, 3771–3775. (cited on page 89)

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. (cited on page 18)

Mirza, M. and Osindero, S., 2014. Conditional generative adversarial nets. *arXiv:1411.1784*, (2014). (cited on page 29)

Mitchell, M.; Aguilar, J.; Wilson, T.; and Van Durme, B., 2013. Open Domain Targeted Sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1643–1654. Association for Computational Linguistics, Seattle, Washington, USA. http://www.aclweb.org/anthology/D13-1171. (cited on pages 88, 89, 90, and 91)

Moeslund, T. B.; Hilton, A.; and Krüger, V., 2006. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104, 2-3 (2006), 90–126. (cited on page 3)

Mun, J.; Cho, M.; and Han, B., 2020. Local-Global Video-Text Interactions for Temporal Grounding. *arXiv:2004.07514 [cs]*, (Apr. 2020). (cited on pages 61 and 73)

Nair, V. and Hinton, G. E., 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814. (cited on page 16)

Nivre, J.; De Marneffe, M.-C.; Ginter, F.; Goldberg, Y.; Hajic, J.; Manning, C. D.; McDonald, R.; Petrov, S.; Pyysalo, S.; Silveira, N.; et al., 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 1659–1666. (cited on page 106)

Nivre, J.; Hall, J.; Nilsson, J.; Chanev, A.; Eryigit, G.; Kübler, S.; Marinov, S.; and Marsi, E., 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13, 2 (2007), 95–135. (cited on page 106)

Park, S.; Shim, H. S.; Chatterjee, M.; Sagae, K.; and Morency, L.-P., 2014. Computational Analysis of Persuasiveness in Social Multimedia: A Novel Dataset and Multimodal Prediction Approach. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, 50–57. ACM, New York, NY, USA. doi:10.1145/2663204.2663260. (cited on pages 6 and 89)

Pech-Pacheco, J. L.; Cristóbal, G.; Chamorro-Martinez, J.; and Fernández-Valdivia, J., 2000. Diatom autofocusing in brightfield microscopy: a comparative study. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 3, 314–317. IEEE. (cited on page 65)

Pellegrini, S.; Ess, A.; Schindler, K.; and Van Gool, L., 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*. (cited on page 27)

Pennington, J.; Socher, R.; and Manning, C., 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543. (cited on pages 18, 46, 64, and 93)

Pham, H.; Manzini, T.; Liang, P. P.; and Poczós, B., 2018. Seq2seq2sentiment: Multimodal Sequence to Sequence Models for Sentiment Analysis. In *Proceedings of Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML)*, 53–63. Association for Computational Linguistics, Melbourne, Australia. doi:10.18653/v1/W18-3308. http://aclweb.org/anthology/W18-3308. (cited on page 90)

Piedimonte, A.; Woods, A. J.; and Chatterjee, A., 2015. Disambiguating ambiguous motion perception: what are the cues? *Frontiers in Psychology*, 6 (2015), 902. doi:10.3389/fpsyg.2015.00902. https://www.frontiersin.org/article/10.3389/fpsyg.2015.00902. (cited on page 23)

Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; and Manandhar, S., 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 27–35. Association for Computational Linguistics and Dublin City University, Dublin, Ireland. http://www.aclweb.org/anthology/S14-2004. (cited on page 88)

Poria, S.; Cambria, E.; and Gelbukh, A., 2015. Deep Convolutional Neural Network Textual Features and Multiple Kernel Learning for Utterance-level Multimodal Sentiment Analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2539–2544. Association for Computational Linguistics, Lisbon, Portugal. http://aclweb.org/anthology/D15-1303. (cited on page 88)

Poria, S.; Cambria, E.; Hazarika, D.; Majumder, N.; Zadeh, A.; and Morency, L.-P., 2017. Context-Dependent Sentiment Analysis in User-Generated Videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 873–883. Association for Computational Linguistics, Vancouver, Canada. doi:10.18653/v1/P17-1081. http://aclweb.org/anthology/P17-1081. (cited on page 90)

Qiu, L.; Zhao, Y.; Shi, W.; Liang, Y.; Shi, F.; Yuan, T.; Yu, Z.; and Zhu, S.-C., 2020. Structured attention for unsupervised dialogue structure induction. *EMNLP*, (2020). (cited on page 106)

Radford, A.; Metz, L.; and Chintala, S., 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, (2016). (cited on page 29)

Ren, S.; He, K.; Girshick, R.; and Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*. (cited on page 65)

Richard, A.; Kuehne, H.; Iqbal, A.; and Gall, J., 2018. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2. (cited on pages 4 and 40)

Rodriguez-Opazo, C.; Fernando, B.; and Li, H., 2018. Action anticipation by predicting future dynamic images. In *The European Conference on Computer Vision (ECCV) Workshops*. (cited on page 9)

Rodriguez-Opazo, C.; Marrese-Taylor, E.; Fernando, B.; Gould, S.; and Li, H., 2021. Dori: Discovering object relationships for moment localization of a natural language query in a video. *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, (2021). (cited on page 9)

Rodriguez-Opazo, C.; Marrese-Taylor, E.; Saleh, F. S.; Li, H.; and Gould, S., 2020. Proposal-free temporal moment localization of a natural-language query in video using guided attention. *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, (2020). (cited on pages 9, 61, 73, and 74)

Rohrbach, A.; Rohrbach, M.; Qiu, W.; Friedrich, A.; Pinkal, M.; and Schiele, B., 2014. Coherent multi-sentence video description with variable level of detail. In *Pattern Recognition*. (cited on pages 21, 41, 48, 60, and 68)

Rohrbach, A.; Rohrbach, M.; Tandon, N.; and Schiele, B., 2015. A Dataset for Movie Description. In *Proceedings of the 2015 conference in computer vision and patter recognition*, 3202–3212. (cited on page 42)

Rohrbach, M.; Regneri, M.; Andriluka, M.; Amin, S.; Pinkal, M.; and Schiele, B., 2012. Script data for attribute-based recognition of composite activities. In *European Conference on Computer Vision*, 144–157. Springer. (cited on page 48)

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J., 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science. (cited on page 17)

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al., 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115, 3 (2015), 211–252. (cited on page 17)

Ryoo, M.; Chen, C.-C.; Aggarwal, J.; and Roy-Chowdhury, A., 2010. An overview of contest on semantic description of human activities (sdha) 2010. In *ICPR*. (cited on page 35)

Ryoo, M. S., 2011. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*. (cited on pages 3, 24, 26, and 35)

Ryoo, M. S. and Aggarwal, J. K., 2010. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html. (cited on pages 21 and 32)

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X., 2016. Improved techniques for training gans. In *Advances in neural information processing systems*, 2234–2242. (cited on page 47)

Severyn, A.; Moschitti, A.; Uryupina, O.; Plank, B.; and Filippova, K., 2014. Opinion Mining on YouTube. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1252–1261. Association for Computational Linguistics, Baltimore, Maryland. doi:10.3115/v1/P14-1118. http://aclweb.org/anthology/P14-1118. (cited on page 90)

Shou, Z.; Wang, D.; and Chang, S.-F., 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 42 and 61)

Siersdorfer, S.; Chelaru, S.; Nejdl, W.; and San Pedro, J., 2010. How useful are your comments?: Analyzing and predicting youtube comments and comment ratings. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10 (Raleigh, North Carolina, USA, 2010), 891–900. ACM, New York, NY, USA. doi:10.1145/1772690.1772781. http://doi.acm.org/10.1145/1772690.1772781. (cited on page 90)

Sigurdsson, G. A.; Russakovsky, O.; and Gupta, A., 2017. What actions are needed for understanding human actions in videos? In *ICCV*. (cited on pages 41, 47, 60, and 63)

Sigurdsson, G. A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; and Gupta, A., 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*. (cited on pages 21 and 48)

Simonyan, K. and Zisserman, A., 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, 568–576. (cited on pages 11, 12, 24, and 42)

Singh, B.; Marks, T. K.; Jones, M.; Tuzel, O.; and Shao, M., 2016. A Multi-Stream Bi-Directional Recurrent Neural Network for Fine-Grained Action Detection. In *CVPR*, 1961–1970. http://openaccess.thecvf.com/content_cvpr_2016/html/Singh_A_Multi-Stream_Bi-Directional_CVPR_2016_paper.html. (cited on page 42)

Singh, G.; Saha, S.; Sapienza, M.; Torr, P. H. S.; and Cuzzolin, F., 2017. Online real-time multiple spatiotemporal action localisation and prediction. In *ICCV*. (cited on pages 27, 32, 35, and 36)

Smola, A. J. and Schölkopf, B., 2004. A tutorial on support vector regression. *Statistics and computing*, 14, 3 (2004), 199–222. (cited on page 13)

Sohn, K.; Lee, H.; and Yan, X., 2015. Learning structured output representation using deep conditional generative models. In *NIPS*. (cited on page 29)

Soomro, K.; Idrees, H.; and Shah, M., 2016a. Online localization and prediction of actions and interactions. *arXiv:1612.01194*, (2016). (cited on pages 3, 24, 26, 35, and 36)

Soomro, K.; Idrees, H.; and Shah, M., 2016b. Predicting the where and what of actors and actions through online action localization. In *CVPR*. (cited on pages 3, 24, 26, 32, 35, and 36)

Soomro, K.; Zamir, A. R.; and Shah, M., 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, (2012). (cited on pages 21 and 32)

Specia, L.; Frank, S.; Sima'an, K.; and Elliott, D., 2016. A Shared Task on Multimodal Machine Translation and Crosslingual Image Description. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, 543–553. Association for Computational Linguistics, Berlin, Germany. doi:10.18653/v1/W16-2346. http://aclweb.org/anthology/W16-2346. (cited on page 88)

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A., 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*. (cited on pages 30 and 33)

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9. (cited on pages 45, 65, and 97)

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z., 2016. Rethinking the Inception Architecture for Computer Vision. 2818–2826. https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.html. (cited on page 47)

Tahara, Y.; Tago, A.; Nakagawa, H.; and Ohsuga, A., 2010. Nicoscene: Video scene search by keywords based on social annotation. In *Active Media Technology* (Eds. A. An; P. Lingras; S. Petty; and R. Huang), vol. 6335 of *Lecture Notes in Computer Science*, 461–474. Springer Berlin Heidelberg. ISBN 978-3-642-15469-0. (cited on page 90)

Tang, D.; Qin, B.; Feng, X.; and Liu, T., 2016a. Effective LSTMs for Target-Dependent Sentiment Classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 3298–3307. The COLING 2016 Organizing Committee, Osaka, Japan. http://aclweb.org/anthology/C16-1311. (cited on page 89)

Tang, D.; Qin, B.; and Liu, T., 2016b. Aspect Level Sentiment Classification with Deep Memory Network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 214–224. Association for Computational Linguistics, Austin, Texas. https://aclweb.org/anthology/D16-1021. (cited on page 89)

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M., 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 4489–4497. (cited on pages 17, 42, 45, and 97)

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*, 5998–6008. (cited on pages 46 and 64)

Villani, V.; Pini, F.; Leali, F.; and Secchi, C., 2018. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 55 (2018), 248–266. (cited on page 105)

Vondrick, C.; Pirsiavash, H.; and Torralba, A., 2016. Anticipating visual representations from unlabeled video. In *CVPR*. (cited on pages 26 and 27)

Wang, J.; Ma, L.; and Jiang, W., 2020. Temporally grounding language queries in videos by contextual boundary-aware prediction. *AAAI*, (2020). (cited on page 72)

Wang, W.; Huang, Y.; and Wang, L., 2019. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 334–343. (cited on page 51)

Wu, B.; Zhong, E.; Tan, B.; Horner, A.; and Yang, Q., 2014. Crowdsourced time-sync video tagging using temporal and personalized topic modeling. In *Proceedings* *of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14 (New York, New York, USA, 2014), 721–730. ACM, New York, NY, USA. doi:10.1145/2623330.2623625. http://doi.acm.org/10.1145/2623330.2623625. (cited on page 90)

Xu, H.; Das, A.; and Saenko, K., 2017. R-c3d: Region convolutional 3d network for temporal activity detection. *ICCV*, (2017). (cited on page 40)

Xu, H.; He, K.; Sigal, L.; Sclaroff, S.; and Saenko, K., 2019. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*. (cited on pages 4, 43, 51, 52, 61, 93, and 94)

Xu, L.; Liu, K.; and Zhao, J., 2014. Joint Opinion Relation Detection Using One-Class Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 677–687. Dublin City University and Association for Computational Linguistics, Dublin, Ireland. http://www.aclweb.org/anthology/C14-1064. (cited on page 90)

Xu, L.-Q. and Li, Y., 2003. Video classification using spatial-temporal features and pca. In *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*, vol. 3, III–485. IEEE. (cited on page 11)

Yang, J.; Liang, S.; and Zhang, Y., 2018. Design Challenges and Misconceptions in Neural Sequence Labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, 3879–3889. Association for Computational Linguistics, Santa Fe, New Mexico, USA. (cited on page 94)

Yang, J. and Zhang, Y., 2018. NCRF++: An Open-source Neural Sequence Labeling Toolkit. In *Proceedings of ACL 2018, System Demonstrations*, 74–79. Association for Computational Linguistics, Melbourne, Australia. doi:10.18653/v1/P18-4013. (cited on page 94)

Yilmaz, A. and Shah, M., 2005. Actions sketch: A novel action representation. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 984–989. IEEE. (cited on page 11)

Yu, G.; Yuan, J.; and Liu, Z., 2012. Predicting human activities using spatio-temporal structure of interest points. In *ACMMM*. (cited on page 26)

Yu, L.; Lin, Z.; Shen, X.; Yang, J.; Lu, X.; Bansal, M.; and Berg, T. L., 2018. MAttNet: Modular Attention Network for Referring Expression Comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1307–1315. (cited on page 62)

Yuan, Y.; Mei, T.; and Zhu, W., 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. *AAAI*, (2019). (cited on pages 43, 51, 52, 61, and 73)

Zadeh, A.; Chen, M.; Poria, S.; Cambria, E.; and Morency, L.-P., 2017. Tensor Fusion Network for Multimodal Sentiment Analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1103–1114. Association for Computational Linguistics, Copenhagen, Denmark. doi:10.18653/v1/D17-1115. http://aclweb.org/anthology/D17-1115. (cited on page 90)

Zadeh, A.; Zellers, R.; Pincus, E.; and Morency, L.-P., 2016. MOSI: Multimodal Corpus of Sentiment Intensity and Subjectivity Analysis in Online Opinion Videos. *arXiv:1606.06259 [cs]*, (Jun. 2016). ArXiv: 1606.06259. (cited on pages 88 and 90)

Zeng, R.; Huang, W.; Tan, M.; Rong, Y.; Zhao, P.; Huang, J.; and Gan, C., 2019. Graph Convolutional Networks for Temporal Action Localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 7094–7103. (cited on page 62)

Zeng, R.; Xu, H.; Huang, W.; Chen, P.; Tan, M.; and Gan, C., 2020. Dense Regression Network for Video Grounding. *arXiv:2004.03545 [cs]*, (Apr. 2020). (cited on page 61)

Zhang, D.; Dai, X.; Wang, X.; Wang, Y.-F.; and Davis, L. S., 2019. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. *CVPR*, (2019). (cited on pages 43, 46, 51, 62, and 72)

Zhang, M.; Zhang, Y.; and Vo, D. T., 2015. Neural Networks for Open Domain Targeted Sentiment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 612–621. Association for Computational Linguistics, Lisbon, Portugal. http://aclweb.org/anthology/D15-1073. (cited on pages 88, 89, 90, 91, and 94)

Zhang, Z.; Zhao, Z.; Zhao, Y.; Wang, Q.; Liu, H.; and Gao, L., 2020. Where Does It Exist: Spatio-Temporal Video Grounding for Multi-Form Sentences. *arXiv:2001.06891 [cs]*, (Jan. 2020). (cited on page 62)

Zhao, X.; Jiang, J.; Yan, H.; and Li, X., 2010. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 56–65. Association for Computational Linguistics, Cambridge, MA. https://www.aclweb.org/anthology/D10-1006. (cited on page 95)

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D., 2017. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2914–2923. (cited on page 40)

Zhou, L.; Louis, N.; and Corso, J. J., 2018a. Weakly-supervised video object grounding from text by loss weighting and object interaction. In *BMVC*. (cited on pages 60 and 68)

Zhou, L.; Xu, C.; and Corso, J. J., 2018b. Towards automatic learning of procedures from web instructional videos. In *AAAI Conference on Artificial Intelligence*, 7590–7598. (cited on pages 21, 60, 68, and 69)

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232. (cited on page 105)

Zuo, Y.; Wu, J.; Zhang, H.; Wang, D.; and Xu, K., 2018. Complementary aspect-based opinion mining. *IEEE Transactions on Knowledge and Data Engineering*, 30, 2 (Feb 2018), 249–262. doi:10.1109/TKDE.2017.2764084. (cited on page 95)