Towards Human Action Understanding in Social Media Videos Using Multimodal Models

by

Oana Ignat

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2022

Doctoral Committee:

Professor Rada Mihalcea, Chair
Research Scientist Fabian Caba Heilbron
Professor Joyce Y. Chai
Assistant Professor David Fouhey
Assistant Professor Andrew Owens

Oana Ignat

oignat@umich.edu

ORCID iD: 0000-0003-0272-5147

ACKNOWLEDGMENTS

Thank you to my parents who dedicated their life to raising me well, always making sure I had everything I needed to learn and develop to be a good human. Thank you for encouraging me to become independent and pursue my dreams. Finishing this thesis is one of those dreams and I am very grateful for your constant support.

Thank you to my advisor, Rada, for taking a chance on me and offering me the beautiful opportunity of pursuing a Ph.D. at a great University. You inspired me with your ambition, optimism and kindness. I learned so much from you, not only about how to conduct research but also how to be a better person, to encourage others to pursue Computer Science. I admire your devotion to open up science for everyone and I look forward to continuing your mission.

Thank you to my thesis committee Rada Mihalcea, Fabian Caba Heilbron, Joyce Chai, David Fouhey and Andrew Owens for your valuable suggestions and support while preparing this dissertation.

Thank you to my co-authors and mentees Weiji Li, Hanwen Miao, Yuhang Zhou and Jiajun Bao. I greatly appreciate your help, dedication and hard work, all of which contributed to my thesis completion.

Thank you to my LIT lab: Steve Wilson, Allie Lahnala, Charlie Welch, Laura Biester, Laura Burdick, MeiXing Dong, Michalis Papakostas, Ian Stewart, Mahmmoud Azab, Veronica Perez-Rosas, Santiago Castro, Ashkan Kazemi, Artem Abzaliev, Naihao Deng, Siqi Shen, Dojune Min, Andrew Lee. Our lab meetings and social events were amazing multicultural experiences and I am grateful for your friendship.

Thank you to my research internships mentors and co-authors Alon Halevy, Y-Lan Boureau, Jane Yu, Paco Guzman, Jean Maillard and Vishrav Chaudhary for your help and guidance that lead to very interesting research projects.

Thank you to the University of Michigan professors, especially to Christine Feak, who taught me how to become a better academic writer and enjoy the process.

Thank you to University of Michigan CRLT-ENGIN team, especially to Audra Baleisis, for the great teaching resources and caring, thoughtful discussions.

Thank you to my close friends whom I met at University of Michigan (Alexandra Veliche, Laura Burdick, Allie Lahnala, Charlie Welch, Victoria Florence, Laura Biester, Michalis Papakostas) for being part of my life, during both happy and difficult times.

Thank you to Santiago Castro for his constant support and encouragement throughout my Ph.D. journey. We shared many research discussions, papers and conferences together and your optimism, calm and humour energized and inspired me.

Thank you to my mentor, Cosmin Lazar, who introduced me to Computer Science research, patiently and clearly answering all my questions. I decided to follow a career in research and pursue a Ph.D. due to our project, my undergraduate thesis, where I discovered how to combine Computer Science and Math in an interesting and fun way.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Human action understanding is one of the most impactful and challenging tasks a computer system can do. Once a computer system learns how to interact with humans, it can assist us in our everyday life activities and significantly improve our quality of life.

Despite the attention it has received in fields such as Natural Language Processing and Computer Vision, and the significant strides towards accurate and robust action recognition and localization systems, human action understanding still remains an unsolved problem.

In this thesis, we introduce and analyze how models can learn from multimodal data, i.e, from what humans *say* and *do* while performing their everyday activities. As a step towards endowing systems with a richer understanding of human actions in online videos, this thesis proposes new techniques that rely on the vision and language channels to address four important challenges: i) human action visibility identification in online videos, ii) temporal human action localization in online videos, iii) human action reason identification in online videos, and iv) human action co-occurrence identification.

We focus on the widely spread genre of lifestyle vlogs, which consist of videos of people performing actions while verbally describing them. We construct a dataset with crowdsourced manual annotations of visible actions, temporal action localization and action reason identification in online vlogs.

We propose a multimodal unsupervised model to automatically infer the reasons corresponding to an action presented in the video, a simple yet effective method to localize the narrated actions based on their expected duration, and a multimodal supervised classification model of action visibility in videos. We also perform ablations on how each modality contributes to solving the tasks and compare the multimodal models performance with the single-modalities models based on the visual content and vlog transcripts.

Finally, we present an extensive analysis of this data, which allows for a better understanding of how the language and visual modalities interact throughout the videos and pave the road for rich avenues for future work.

CHAPTER 1

Introduction

The goal of technology is to improve human life. One way this can be achieved is through automatizing repetitive, mundane or dangerous activities. In manufacturing factories, most repetitive actions are being automatized by programming robots to perform those actions [1]. There is also a need for robots assisting with everyday life activities (e.g., cleaning, cooking), especially for elderly or disabled people. Automatizing mundane activities would result in having more time for rest, well-being and creative activities. For instance, there have been attempts to program kitchen robots to make food, but with little to no success [2]. This is because everyday human actions are very complex and widely diverse. In order to make progress on this task, computer models need a strong grasp of human action understanding.

Understanding human actions means that computers need to recognize our actions, identify why we choose to perform each action, and predict what actions we intend to do next. Human action understanding covers multiple complex tasks such as human action recognition, temporal localization (i.e., when the action happens) and action causal reasoning (i.e., why the action is performed).

Over the past decades, human action understanding has been studied extensively in the Computer Vision and Natural Language Processing research communities. Textual and visual information complement each other. From visual data, computers learn fine-grained information about the human actions, such as human pose [3] and objects [4, 5] used to perform the action. From textual data, computers learn more high-level, social information such as motivation [6], cause-effect [7] or human values [8, 9]. For example, from a video a computer can learn about how the *running* activity looks like (i.e., how the legs and arms move), while from text (i.e., someone's tweet about how they feel refreshed and less anxious after a run) a computer can learn about its benefits and why the person will probably continue to perform the activity. Both sources of information are required for the computer to understand the action of *running*.

Research in Computer Vision has taken significant steps towards robust and accurate action recognition and localization systems. Human action understanding covers many research topics in computer vision, including human activity recognition in video [10, 11, 12], human pose estimation [3, 13] and temporal human action localization in videos [14, 15]. Visual datasets with focus on specific domains have been produced, among which most popular have been sports [16, 17], cooking [18, 19, 20, 21] and instructional videos [22, 23].

Natural Language Processing helps empower machines to understand human language [24]. Computational approaches have been applied for extracting and representing human activities from text data with the goal of understanding and modeling human behaviour [25, 26, 27]. Datasets structured and categorized textual data into action motivations [6], cause-effect relations [7, 28] and hierarchies of verbs/actions [29].

Despite this extensive research, human action understanding remains an unsolved problem. This is due to the many challenges that models have to overcome both in the visual domain (i.e., camera viewpoint variation, lighting, changes in scale, background clutter or partial occlusions) [30] as well as in the textual domain (i.e., semantic and syntactic ambiguity [31] or parsing errors [32])

To overcome these challenges, the deep learning community started constructing increasingly complex datasets that are closer to real life scenarios. This is often done by leveraging the unlimited web data by collecting millions of videos of human actions from social media platforms like YouTube [33, 34, 4, 22, 35]. These datasets are open ended (i.e., new videos are being uploaded to YouTube every second) and challenging due to diverse range of actions, filming perspectives and illumination conditions. Among these datasets, the predefined action classes present in [33, 34] are replaced by natural language queries, collected from the video transcripts [22, 35]. This further increases the complexity of the action understanding task as one can describe the same actions in multiple different ways: e.g., the predefined action class "eat" can be expressed in different natural language queries as "grab a bite" or "get a snack". This leads to further challenges in solving the more complex, but realistic scenarios.

To address some of these challenges, in this thesis, we explore how we can use multimodal (textual and visual) information from online narrated videos, to enable automatic models to learn about human actions. In particular, we build datasets and models for automatic human action detection, localization, co-occurrence identification and causal reasoning. Using machine learning techniques applied on challenging lifestyle vlogs from YouTube, this thesis provides empirical evidence that models can learn about human actions from multimodal data.

The goal of the research is to introduce and analyze how models can learn from multimodal data, i.e from what humans say and do while performing their everyday activities.

Specifically, the thesis seeks to answer the following research questions:

1. **Are vlogs well suited for learning about human actions and behaviors?**

   Previous work started by searching videos on YouTube using keywords that describe an action [33, 36, 37]. Because of the "boring/mainstream" nature of most routine actions, searching for them directly returned few results or returned unexpected videos. For example, searching for videos with the query "drinking tea" results mainly in unusual videos such as dogs or birds drinking tea. This issue can be addressed by paying people to act out everyday scenarios [38], but this can be very expensive. In our work, we address this problem by changing the approach used to search for videos. Instead of searching for actions in an explicit way, using queries such as "opening a fridge" or "making the bed", we search for more general videos using queries such as "my morning routine."

   In Chapters 2, 3, 4 and 5 we present qualitative and quantitative analyses of the videos, which indicate that they are well suited for learning about human actions and behaviours.

   In Chapter 2 we show that lifestyle vlogs contain rich transcripts, a high variety of human actions and are very popular, with the potential to become a very large and actively growing data source. Given the prevalence of vlogs in online platforms, automatically extracting action names from their transcripts can lead to large-scale inexpensive action annotation.

   Moreover, vlogs typically include transcripts with complex natural language expressions, which allow us to find an alternative to the costly process of manual annotations. In Chapter 4 we show that lifestyle vlogs can be a source of commonsense knowledge about human activities, due to how vloggers verbally express their intentions and feelings about the activities they perform.

   Finally, vlogs contain temporally overlapping actions (Chapter 3) or actions that tend to co-occur in the same interval of time (Chapter 5), providing rich information about the interconnection of human actions.

2. **Can machine learning models learn useful characteristics of human actions from lifestyle vlogs?**

   In Chapters 2, 3, 4 and 5 we show that machine learning models can learn useful characteristics about human actions from lifestyle vlogs. We collect and annotate

datasets based on the textual and visual information from lifestyle vlogs and train machine learning models for complex tasks on these datasets.

Lifestyle vlogs present a person's everyday routine in which the vlogger visually records the activities they perform during a normal day and verbally expresses their intentions and feelings about those activities.

In Chapter 2 we show that routine videos contain a very diverse set of activities, from waking up in the morning and taking a shower, to working out and making a meal. This diversity of actions in one video translates to many more diverse filming perspectives in the same video, which presents a novel challenge for action understanding models.

The routine nature of the videos makes them a novel and valuable data source for learning about the temporal connections between human actions (Chapter 5) and how they can be used for building stronger action localization and action prediction models (Chapter 3).

Lifestyle vlogs contain a wide variety of actions that are more akin to real-life settings, such as "grab my Kindle", "do some reading", or "chill out". The rich videos and transcripts can enable machine learning models to learn not only about how to perform an action (Chapter 2) but also why the vlogger chooses to perform it (Chapter 4).

Because of these characteristics, lifestyle vlogs are a rich data source for an in depth study of human actions and behaviors.

3. **Are multimodal models more effective than uni-modal models in solving the tasks and if so, how to combine different modalities?**

   Learning the connections between vision and language is essential to human action understanding.

   We address this question in Chapters 2, 3, 4 and 5, where we introduce multimodal neural architectures that combine information drawn from visual and linguistic clues, and show that it improves over models that rely on one modality at a time.

   We explore different ways of combining the text and video information, to make sure the models can leverage each input. This is known to be a difficult task, as most models can easily exploit biases in our language in order to "solve" the task [39, 40]. This results in models that ignore visual information, leading to an inflated sense of their capability [41]. To understand how much each modality contributes to solving

the task, we also perform ablation studies where we breakdown the performance of models for each modality.

For action visibility classification (Chapter 2), we build a multimodal model where we concatenate the textual and visual representations and input them through a three-layer feed forward network. For action localization (Chapter 3), we build a multimodal model by using the MPU [14] model (vector element-wise addition, vector element-wise multiplication and vector concatenation followed by a Fully Connected layer) for long actions and the video transcript timestamp for short actions. For action reason identification (Chapter 4), we combine the information from both textual and visual modalities, by using a T5 [42] encoder-decoder model. The text input is passed through an embedding layer, while the video features are passed through a linear layer. For action co-occurrence identification (Chapter 5), we build a multimodal model using various data representations: textual, visual and graph topology information. We concatenate all the representations and input them in an SVM [43] classifier.

4. **Can we build automatic models for solving physical tasks related to human action understanding such as action visibility classification and action localization?**

   Physical tasks represent an important part of action understanding, that rely mostly on leveraging visual information: i.e., tasks such as detecting if an action appears in the video or where in the video an action is localized.

   We address this question in Chapters 2 and 3, where we formalize the tasks of action visibility classification and action temporal localization in online vlogs. In Chapter 2 we introduce a novel dataset consisting of short video clips paired with sets of actions mentioned in the video transcripts, as well as manual annotations of whether the actions are visible in the video. In Chapter 3 we extend that dataset with information about where the visible actions are localized in the videos.

   We also implement and test automatic models for action visibility classification (Chapter 2) and action temporal localization (Chapter 3).

5. **Can we build automatic models for solving commonsense tasks related to human action understanding such as action reason classification and action co-occurrence identification?**

   Commonsense tasks for action understanding represent another important part of action understanding, that require leveraging both visual and context information [6, 7]:

i.e., tasks such as detecting why the action is performed or is two actions usually happen in the same interval of time in a video.

We address this question in Chapters 4 and 5, where we formalize the new tasks of action reason identification and action co-occurrence identification in online vlogs. In Chapter 4 we introduce a new dataset consisting of (action, context, reasons) tuples manually labeled in online vlogs, covering actions and their reasons drawn from ConceptNet [6] as well as crowdsourcing contributions. In Chapter 5 we introduce a new dataset consisting of a large graph of co-occurring visual actions and their corresponding video-clips in online vlogs.

We also propose several models to solve the tasks of human action reason identification (Chapter 4) and action co-occurrence identification (Chapter 5).

1.1 Thesis Organization

The thesis is organized as follows. In Chapter 2, we introduce the task of identifying human actions visible in online videos and propose a multimodal model that leverages information derived from visual and linguistic clues to automatically infer which actions are visible in a video. In Chapter 3, we consider the task of temporal human action localization in lifestyle vlogs and propose a simple yet effective method to localize the narrated actions based on their expected duration. In Chapter 4, we aim to automatically identify human action reasons in online videos and describe a multimodal model that leverages visual and textual information to automatically infer the reasons corresponding to an action presented in the video. In Chapter 5, we aim to automatically identify human actions co-occurrence in lifestyle vlogs and describe models that leverage textual, visual, and graph information to solve the action co-occurrence identification task as a link prediction task.

Finally, Chapter 6 summarizes the findings of the thesis and revisits the research questions posed in the introduction, also highlighting the contributions made by this thesis in the field of multimodal human action understanding in online videos.

1.2 Funding Acknowledgments

The work in this dissertation is based in part upon work supported by the Michigan Institute for Data Science, by the National Science Foundation (grant #1815291), by the John Templeton Foundation (grant #61156), by DARPA (grant #HR001117S0026-AIDA-FP-045) and by a grant from the Automotive Research Center (ARC) at the University of Michigan.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of these organizations.

CHAPTER 2

Human Action Visibility Classification

2.1 Introduction

There has been a surge of recent interest in detecting human actions in videos. Work in this space has mainly focused on learning actions from articulated human pose [46, 47, 48] or mining spatial and temporal information from videos [49, 50]. A number of resources have been produced, including Action Bank [51], NTU RGB+D [52], SBU Kinect Interaction [53], and PKU-MMD [54].

Most research on video action detection has gathered video information for a set of predefined actions [33, 36, 37], an approach known as *explicit data gathering* [4]. For instance, given an action such as "open door," a system would identify videos that include a visual depiction of this action. While this approach is able to detect a specific set of actions, whose choice may be guided by downstream applications, it achieves high precision at the cost of low recall. In many cases, the set of predefined actions is small (e.g., 203 activity classes in Heilbron et al. 33), and for some actions, the number of visual depictions is very small.

An alternative approach is to start with a set of videos, and identify all the actions present in these videos [55, 56]. This approach has been referred to as *implicit data gathering,* and it typically leads to the identification of a larger number of actions, possibly with a small number of examples per action.

In this work, we use an implicit data gathering approach to label human activities in videos. To the best of our knowledge, we are the first to explore video action recognition using both transcribed audio and video information. We focus on the popular genre of lifestyle vlogs, which consist of videos of people demonstrating routine actions while verbally describing them. We use these videos to develop methods to identify if actions are visually present.

The work makes three main contributions. First, we introduce a novel dataset consisting

Dataset	#Actions	#Verbs	#Actors	Implicit	Label types
Ours	4340	580	10	✓	✓
VLOG [4]	-	-	10.7k	✓	✓
Kinetics [37]	600	270	-	x	x
ActivityNet [33]	203	-	-	x	x
MIT [44]	339	339	-	x	x
AVA [45]	80	80	192	✓	x
Charades [38]	157	30	267	x	x
MPII Cooking [18]	78	78	12	✓	x

Table 2.1: Comparison between our dataset and other video human action recognition datasets. # Actions show either the number of action classes in that dataset (for the other datasets), or the number of unique visible actions in that dataset (ours); # Verbs shows the number of unique verbs in the actions; Implicit is the type of data gathering method (versus explicit); Label types are either post-defined (first gathering data and then annotating actions): ✓, or pre-defined (annotating actions before gathering data): x.

of 1,268 short video clips paired with sets of actions mentioned in the video transcripts, as well as manual annotations of whether the actions are visible or not. The dataset includes a total of 14,769 actions, 4,340 of which are visible. Second, we propose a set of strong baselines to determine whether an action is visible or not. Third, we introduce a multimodal neural architecture that combines information drawn from visual and linguistic clues, and show that it improves over models that rely on one modality at a time.

By making progress towards automatic action recognition, in addition to contributing to video understanding, this work has a number of important and exciting applications, including sports analytics [57], human-computer interaction [58], and automatic analysis of surveillance video footage [59].

The chapter is organized as follows. We begin by discussing related work, then describe our data collection and annotation process. We next overview our experimental set-up and introduce a multimodal method for identifying visible actions in videos. Finally, we discuss our results and conclude with general directions for future work.

2.2 Related Work

There has been substantial work on action recognition in the computer vision community, focusing on creating datasets [60, 61, 38, 33] or introducing new methods [62, 11, 63, 64].

Table 2.1 compares our dataset with previous action recognition datasets.[1]

The largest datasets that have been compiled to date are based on YouTube videos [33, 36, 37]. These actions cover a broad range of classes including human-object interactions such as cooking [65, 66, 18] and playing tennis [61], as well as human-human interactions such as shaking hands and hugging [45].

Similar to our work, some of these previous datasets have considered everyday routine actions [33, 36, 37]. However, because these datasets rely on videos uploaded on YouTube, it has been observed they can be potentially biased towards unusual situations [37]. For example, searching for videos with the query "drinking tea" results mainly in unusual videos such as dogs or birds drinking tea. This bias can be addressed by paying people to act out everyday scenarios [38], but this can end up being very expensive. In our work, we address this bias by changing the approach used to search for videos. Instead of searching for actions in an explicit way, using queries such as "opening a fridge" or "making the bed," we search for more general videos using queries such as "my morning routine."

This approach has been referred to as implicit (as opposed to explicit) data gathering, and was shown to result in a greater number of videos with more realistic action depictions [4].

Although we use implicit data gathering as proposed in the past, unlike [4] and other human action recognition datasets, we search for routine videos that contain rich audio descriptions of the actions being performed, and we use this transcribed audio to extract actions. In these lifestyle vlogs, a vlogger typically performs an action while also describing it in detail. To the best of our knowledge, we are the first to build a video action recognition dataset using both transcribed audio and video information.

Another important difference between our methodology and previously proposed methods is that we extract action labels from the transcripts. By gathering data before annotating the actions, our action labels are post-defined (as in Fouhey et al. 4). This is unlike the majority of the existing human action datasets that use pre-defined labels [38, 33, 36, 37, 45, 66, 18, 44]. Post-defined labels allow us to use a larger set of labels, expanding on the simplified label set used in earlier datasets. These action labels are more inline with everyday scenarios, where people often use different names for the same action. For example, when interacting with a robot, a user could refer to an action in a variety of ways; our dataset includes the actions "stick it into the freezer," "freeze it," "pop into the freezer," and "put into the freezer," variations, which would not be included in current human action recognition

[1]Note that the number of actions shown for our dataset reflects the number of unique visible actions in the dataset and not the number of action classes, as in other datasets. This is due to our annotation process (see §2.3).

datasets.

In addition to human action recognition, our work relates to other multimodal tasks such as visual question answering [67, 68], video summarization [69, 70], and mapping text descriptions to video content [71, 72]. Specifically, we use an architecture similar to [67], where an LSTM [73] is used together with frame-level visual features such as Inception [74], and sequence-level features such as C3D [64]. However, unlike [67] who encode the textual information (question-answers pairs) using an LSTM, we chose instead to encode our textual information (action descriptions and their contexts) using a large-scale language model ELMo [75].

Similar to previous research on multimodal methods [76, 77, 78, 67], we also perform feature ablation to determine the role played by each modality in solving the task. Consistent with earlier work, we observe that the textual modality leads to the highest performance across individual modalities, and that the multimodal model combining textual and visual clues has the best overall performance.

2.3 Data Collection and Annotation

We collect a dataset of routine and do-it-yourself (DIY) videos from YouTube, consisting of people performing daily activities, such as making breakfast or cleaning the house. These videos also typically include a detailed verbal description of the actions being depicted. We choose to focus on these lifestyle vlogs because they are very popular, with tens of millions having been uploaded on YouTube; Table 2.2 shows the approximate number of videos available for several routine queries. Vlogs also capture a wide range of everyday activities; on average, we find thirty different visible human actions in five minutes of video.

By collecting routine videos, instead of searching explicitly for actions, we do *implicit* data gathering, a form of data collection introduced by Fouhey et al. 4. Because everyday actions are common and not unusual, searching for them directly does not return many results. In contrast, by collecting routine videos, we find many everyday activities present in these videos.

2.3.1 Data Gathering

We build a data gathering pipeline (see Figure 2.1) to automatically extract and filter videos and their transcripts from YouTube. The input to the pipeline is manually selected YouTube channels. Ten channels are chosen for their rich routine videos, where the actor(s) describe

Query	Results
my morning routine	28M+
my after school routine	13M+
my workout routine	23M+
my cleaning routine	13M+
DIY	78M+

Table 2.2: Approximate number of videos found when searching for routine and do-it-yourself queries on YouTube.

their actions in great detail. From each channel, we manually select two different playlists, and from each playlist, we randomly download ten videos.

The following data processing steps are applied:

Transcript Filtering. Transcripts are automatically generated by YouTube. We filter out videos that do not contain any transcripts or that contain transcripts with an average (over the entire video) of less than 0.5 words per second.

These videos do not contain detailed action descriptions so we cannot effectively leverage textual information.

Extract Candidate Actions from Transcript. Starting with the transcript, we generate a noisy list of potential actions. This is done using the Stanford parser [79] to split the transcript into sentences and identify verb phrases, augmented by a set of hand-crafted rules to eliminate some parsing errors. The resulting actions are noisy, containing phrases such as "found it helpful if you" and "created before up the top you."

Segment Videos into Miniclips. The length of our collected videos varies from two minutes to twenty minutes. To ease the annotation process, we split each video into miniclips (short video sequences of maximum one minute). Miniclips are split to minimize the chance that the same action is shown across multiple miniclips. This is done automatically, based on the transcript timestamp of each action. Because YouTube transcripts have timing information, we are able to line up each action with its corresponding frames in the video. We sometimes notice a gap of several seconds between the time an action occurs in the transcript and the time it is shown in the video. To address this misalignment, we first map the actions to the miniclips using the time information from the transcript. We then expand the miniclip by 15 seconds before the first action and 15 seconds after the last action. This increases the chance that all actions will be captured in the miniclip.

Motion Filtering. We remove miniclips that do not contain much movement. We sample one out of every one hundred frames of the miniclip, and compute the 2D correlation coefficient between these sampled frames. If the median of the obtained values is greater than

a certain threshold (we choose 0.8), we filter out the miniclip.

Videos with low movement tend to show people sitting in front of the camera, describing their routine, but not acting out what they are saying. There can be many actions in the transcript, but if they are not depicted in the video, we cannot leverage the video information.

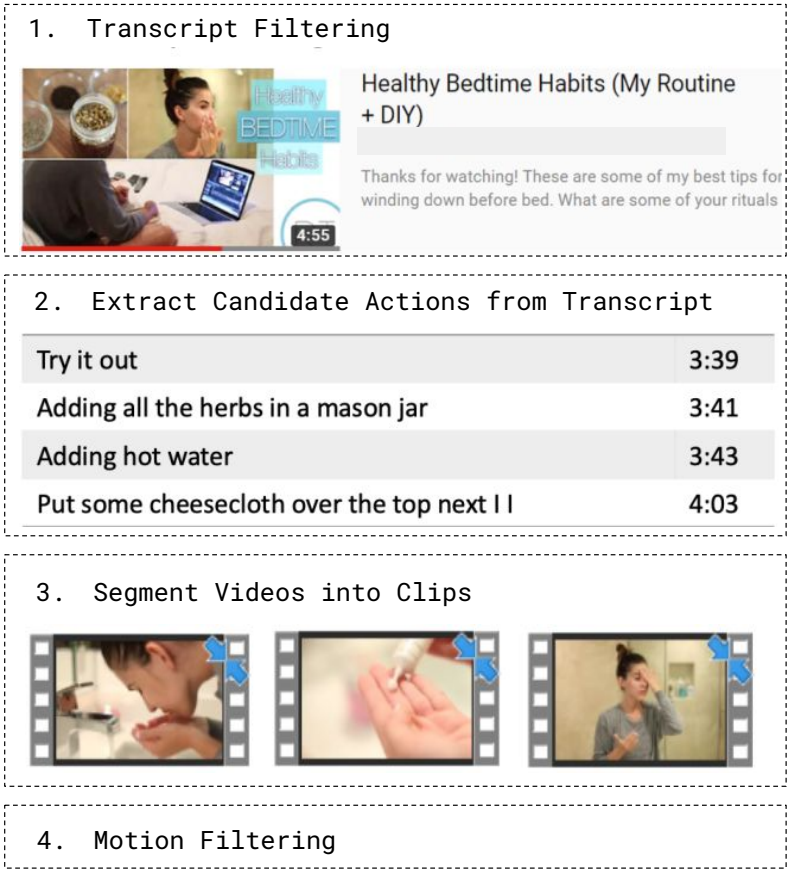


Figure 2.1: Overview of the data gathering pipeline.

2.3.2 Visual Action Annotation

Our goal is to identify which of the actions extracted from the transcripts are visually depicted in the videos. We create an annotation task on Amazon Mechanical Turk (AMT) to identify actions that are visible.

We give each AMT turker a HIT consisting of five miniclips with up to seven actions generated from each miniclip. The turker is asked to assign a label (*visible* in the video; *not visible* in the video; *not an action*) to each action. Figure 2.3 shows the AMT interface used. Because it is difficult to reliably separate *not visible* and *not an action*, we group

...
03:24 you're gonna actually cook it
03:27 and it you're gonna bake it for
03:30 about six hours it's definitely a
03:32 long time so keep in mind that it's
03:34 basically just dehydrating it
03:50 after what seems like an eternity in
03:53 the oven you're going to take it out
03:55 it's actually dehydrated at that point
03:57 which is fabulous because you can
03:59 pull it right off the baking sheet and
04:01 you're going to put it on to some
04:03 parchment paper and then you're
...

Action	Visible?
actually cook it	✓
bake it for	✓
take it out	✓
pull it right off the baking sheet	✓
put it on to some parchment paper	✓
so keep in mind that	x
seems like an eternity in the oven	x
dehydrated at that point which	x

Figure 2.2: Sample video frames, transcript, and annotations.

these labels together.

Each miniclip is annotated by three different turkers. For the final annotation, we use the label assigned by the majority of turkers, i.e., *visible* or *not visible / not an action*.

To help detect spam, we identify and reject the turkers that assign the same label for every action in all five miniclips that they annotate. Additionally, each HIT contains a ground truth miniclip that has been pre-labeled by two reliable annotators. Each ground truth miniclip has more than four actions with labels that were agreed upon by both reliable annotators. We compute accuracy between a turker's answers and the ground truth annotations; if this accuracy is less than 20%, we reject the HIT as spam.

After spam removal, we compute the agreement score between the turkers using Fleiss kappa [80]. Over the entire data set, the Fleiss agreement score is 0.35, indicating fair agreement. On the ground truth data, the Fleiss kappa score is 0.46, indicating moderate agreement. This fair to moderate agreement indicates that the task is difficult, and there are cases where the visibility of the actions is hard to label. To illustrate, Figure 2.4 shows examples where the annotators had low agreement.

Table 2.3 shows statistics for our final dataset of videos labeled with actions, and Figure 2 shows a sample video and transcript, with annotations.

For our experiments, we use the first eight YouTube channels from our dataset as train data, the ninth channel as validation data and the last channel as test data. Statistics for this split are shown in Table 2.4.

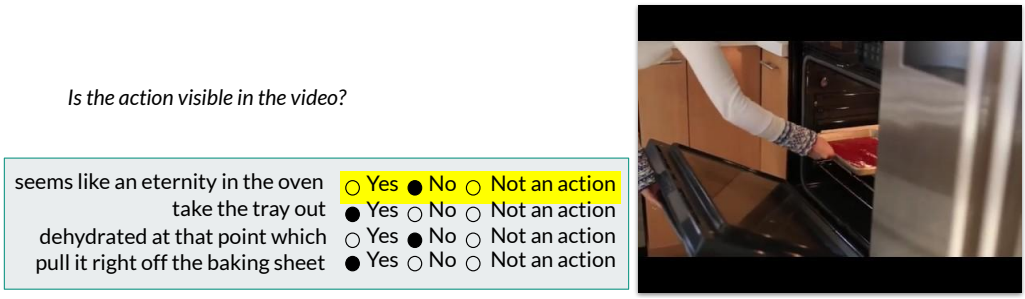


Figure 2.3: Annotation tool used by Amazon Mechanical Turk workers to annotate if an action is visible or not in the video.

Videos	177
Video hours	21
Transcript words	302,316
Miniclips	1,268
Actions	14,769
Visible actions	4,340
Non-visible actions	10,429

Table 2.3: Data statistics.

Action	#1	#2	#3	GT
make sure your skin	x	x	✓	x
cleansed before you	✓	x	✓	✓
do all that	x	x	✓	x
absorbing all that serum when there	x	x	✓	x
move on	x	x	x	x

Figure 2.4: An example of low agreement. The table shows actions and annotations from workers #1, #2, and #3, as well as the ground truth (GT). Labels are: visible - ✓, not visible - x. The bottom row shows screenshots from the video. The Fleiss kappa agreement score is -0.2.

	Train	Test	Validation
# Actions	11,403	1,999	1,367
# Miniclips	997	158	113
# Actions/ Miniclip	11.4	12.6	12.0

Table 2.4: Statistics for the experimental data split.

2.3.3 Discussion

The goal of our dataset is to capture naturally-occurring, routine actions. Because the same action can be identified in different ways (e.g., "pop into the freezer", "stick into the freezer"), our dataset has a complex and diverse set of action labels. These labels demonstrate the language used by humans in everyday scenarios; because of that, we choose not to group our labels into a pre-defined set of actions. Table 2.1 shows the number of unique verbs, which can be considered a lower bound for the number of unique actions in our dataset. On average, a single verb is used in seven action labels, demonstrating the richness of our dataset.

The action labels extracted from the transcript are highly dependent on the performance of the constituency parser. This can introduce noise or ill-defined action labels. Some acions contain extra words (e.g., "brush my teeth of course"), or lack words (e.g., "let me just"). Some of this noise is handled during the annotation process; for example, most actions that lack words are labeled as "not visible" or "not an action" because they are hard to interpret.

2.4 Identifying Visible Actions in Videos

Our goal is to determine if actions mentioned in the transcript of a video are visually represented in the video. We develop a multimodal model that leverages both visual and textual information, and we compare its performance with several single-modality baselines.

2.4.1 Data Processing and Representations

Starting with our annotated dataset, which includes miniclips paired with transcripts and candidate actions drawn from the transcript, we extract several layers of information, which we then use to develop our multimodal model, as well as several baselines.

Action Embeddings. To encode each action, we use both GloVe [81] and ELMo [75] embeddings. When using GloVe embeddings, we represent the action as the average of

all its individual word embeddings. We use embeddings with dimension 50. When using ELMo, we represent the action as a list of words which we feed into the default ELMo embedding layer.[2] This performs a fixed mean pooling of all the contextualized word representations in each action.

Part-of-speech (POS). We use POS information for each action. Similar to word embeddings [81], we train POS embeddings. We run the Stanford POS Tagger [82] on the transcripts and assign a POS to each word in an action. To obtain the POS embeddings, we train GloVe on the Google N-gram corpus[3] using POS information from the five-grams. Finally, for each action, we average together the POS embeddings for all the words in the action to form a POS embedding vector.

Context Embeddings. Context can be helpful to determine if an action is visible or not. We use two types of context information, action-level and sentence-level. Action-level context takes into account the previous action and the next action; we denote it as Context_A. These are each calculated by taking the average of the action's GloVe embeddings. Sentence-level context considers up to five words directly before the action and up to five words after the action (we do not consider words that are not in the same sentence as the action); we denote it as Context_S. Again, we average the GLoVe embeddings of the preceding and following words to get two context vectors.

Concreteness. Our hypothesis is that the concreteness of the words in an action is related to its visibility in a video.

We use a dataset of words with associated concreteness scores from [83]. Each word is labeled by a human annotator with a value between 1 (very abstract) and 5 (very concrete). The percentage of actions from our dataset that have at least one word in the concreteness dataset is 99.8%. For each action, we use the concreteness scores of the verbs and nouns in the action. We consider the concreteness score of an action to be the highest concreteness score of its corresponding verbs and nouns. Table 2.5 shows several sample actions along with their concreteness scores and their visiblity.

Video Representations. We use YOLO9000 [84] to identify objects present in each miniclip. We choose YOLO9000 for its high and diverse number of labels (9,000 unique labels). We sample the miniclips at a rate of 1 frame-per-second, and we use the YOLO9000 model pre-trained on COCO [85] and ImageNet [86].

We represent a video both at the frame level and the sequence level. For frame-level video features, we use the Inception V3 model [74] pre-trained on ImageNet. We extract the

[2]Implemented as the ELMo module in Tensorflow

[3]http://storage.googleapis.com/books/ngrams/books/ datasetsv2.html

Action	Con.	Visible?
cook things in **water**	5.00	✓
head right into my **kitchen**	4.97	✓
throw it into the **washer**	4.70	✓
told you what	2.31	x
share my thoughts	2.96	x
prefer them	1.62	x

Table 2.5: Visible actions with high concreteness scores (Con.), and non-visible actions with low concreteness scores. The noun or verb with the highest concreteness score is in bold.

Action	Visible in the miniclip?
put my son	x
sleep after we	x
done dinner	x
get comfortable	✓
pick out some pajamas	✓
start with my skincare	x
cleanse if I or even	x

output of the very last layer before the Flatten operation (the "bottleneck layer"); we choose this layer because the following fully connected layers are too specialized for the original task they were trained for. We extract Inception V3 features from miniclips sampled at 1 frame-per-second.

For sequence-level video features, we use the C3D model [64] pre-trained on the Sports-1M dataset [61]. Similarly, we take the feature map of the sixth fully connected layer. Because C3D captures motion information, it is important that it is applied on consecutive frames. We take each frame used to extract the Inception features and extract C3D features from the 16 consecutive frames around it.

We use this approach because combining Inception V3 and C3D features has been shown to work well in other video-based models [67, 11, 37].

2.4.2 Baselines

Using the different data representations described in Section 2.4.1, we implement several baselines.

Concreteness. We label as visible all the actions that have a concreteness score above a certain threshold, and label as non-visible the remaining ones. We fine tune the threshold

Action: brush my teeth
Object detected: toothbrush
WUP(brush, toothbrush) = 0.94

Action: chop my vegetables
Object detected: carrot
WUP(vegetables, carrot) = 0.9

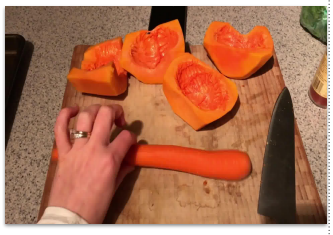


Figure 2.5: Example of frames, corresponding actions, object detected with YOLO, and the object - word pair with the highest WUP similarity score in each frame.

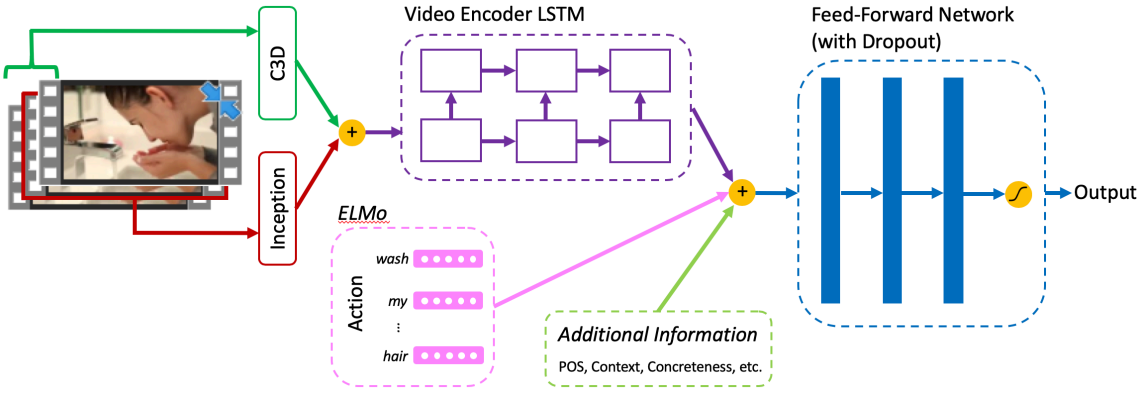


Figure 2.6: Overview of the multimodal neural architecture. + represents concatenation.

on our validation set; for fine tuning, we consider threshold values between 3 and 5. Table 2.6 shows the results obtained for this baseline.

Feature-based Classifier. For our second set of baselines, we run a classifier on subsets of all of our features. We use an SVM [43], and perform five-fold cross-validation across the train and validation sets, fine tuning the hyper-parameters (kernel type, C, gamma) using a grid search. We run experiments with various combinations of features: action GloVe embeddings; POS embeddings; embeddings of sentence-level context (Context_S) and action-level context (Context_A); concreteness score. The combinations that perform best during cross-validation on the combined train and validation sets are shown in Table 2.6.

LSTM and ELMo. We also consider an LSTM model [73] that takes as input the tokenized

action sequences padded to the length of the longest action. These are passed through a trainable embedding layer, initialized with GloVe embeddings, before the LSTM. The LSTM output is then passed through a feed forward network of fully connected layers, each followed by a dropout layer [87] at a rate of 50%. We use a sigmoid activation function after the last hidden layer to get an output probability distribution. We fine tune the model on the validation set for the number of training epochs, batch size, size of LSTM, and number of fully-connected layers.

We build a similar model that embeds actions using ELMo (composed of 2 bi-LSTMs). We pass these embeddings through the same feed forward network and sigmoid activation function. The results for both the LSTM and ELMo models are shown in Table 2.6.

YOLO **Object Detection.** Our final baseline leverages video information from the YOLO9000 object detector. This baseline builds on the intuition that many visible actions involve visible objects. We thus label an action as visible if it contains at least one noun similar to objects detected in its corresponding miniclip. To measure similarity, we compute both the Wu-Palmer (WUP) path-length-based semantic similarity [88] and the cosine similarity on the GloVe word embeddings. For every action in a miniclip, each noun is compared to all detected objects and assigned a similarity score. As in our concreteness baseline, the action is assigned the highest score of its corresponding nouns. We use the validation data to fine tune the similarity threshold that decides if an action is visible or not. The results are reported in Table 2.6. Examples of actions that contain one or more words similar to detected objects by YOLO can be seen in Figure 2.5.

2.5 Multimodal Model

Each of our baselines considers only a single modality, either text or video. While each of these modalities contributes important information, neither of them provides a full picture. The visual modality is inherently necessary, because it shows the visibility of an action. For example, the same spoken action can be labeled as either *visible* or *non-visible*, depending on its visual context; we find 162 unique actions that are labeled as both visible and not visible, depending on the miniclip. This ambiguity has to be captured using video information. However, the textual modality provides important clues that are often missing in the video. The words of the person talking fill in details that many times cannot be inferred from the video. For our full model, we combine both textual and visual information to leverage both modalities.

We propose a multimodal neural architecture that combines encoders for the video and text modalities, as well as additional information (e.g., concreteness). Figure 2.6 shows

our model architecture. The model takes as input a (miniclip m, action a) pair and outputs the probability that action a is visible in miniclip m. We use C3D and Inception V3 video features extracted for each frame, as described in Section 2.4.1. These features are concatenated and run through an LSTM.

To represent the actions, we use ELMo embeddings (see Section 2.4.1). These features are concatenated with the output from the video encoding LSTM, and run through a three-layer feed forward network with dropout. Finally, the result of the last layer is passed through a sigmoid function, which produces a probability distribution indicating whether the action is visible in the miniclip. We use an RMSprop optimizer [89] and fine tune the number of epochs, batch size and size of the LSTM and fully-connected layers.

Method	Input	Accuracy	Precision	Recall	F1
	BASELINES				
Majority	Action	0.692	0.692	1.0	0.81
Threshold	Concreteness	0.685	0.7	0.954	0.807
Feature-based Classifier	Action$_G$	0.715	0.722	0.956	0.823
	Action$_G$, POS	0.701	0.702	**0.986**	0.820
	Action$_G$, Context$_S$	0.725	0.736	0.938	0.825
	Action$_G$, Context$_A$	0.712	0.722	0.949	0.820
	Action$_G$, Concreteness	0.718	0.729	0.942	0.822
	Action$_G$, Context$_S$, Concreteness	**0.728**	**0.742**	0.932	**0.826**
LSTM	Action$_G$	0.706	0.753	0.857	0.802
ELMo	Action$_G$	**0.726**	**0.771**	**0.859**	**0.813**
YOLO	Miniclip	0.625	0.619	0.448	0.520
	MULTIMODAL NEURAL ARCHITECTURE (FIGURE 2.6)				
multimodal Model	Action$_E$, Inception	0.722	0.765	0.863	0.811
	Action$_E$, Inception, C3D	0.725	0.769	0.869	0.814
	Action$_E$, POS, Inception, C3D	0.731	0.763	0.885	0.820
	Action$_E$, Context$_S$, Inception, C3D	0.725	**0.770**	0.859	0.812
	Action$_E$, Context$_A$, Inception, C3D	0.729	0.757	0.895	0.820
	Action$_E$, Concreteness, Inception, C3D	0.723	0.768	0.860	0.811
	Action$_E$, POS, Context$_S$, Concreteness, Inception, C3D	**0.737**	0.758	**0.911**	**0.827**

Table 2.6: Results from baselines and our best multimodal method on validation and test data. Action$_G$ indicates action representation using GloVe embedding, and Action$_E$ indicates action representation using ELMo embedding. Context$_S$ indicates sentence-level context, and Context$_A$ indicates action-level context.

2.6 Evaluation and Results

Table 2.6 shows the results obtained using the multimodal model for different sets of input features. The model that uses all the input features available leads to the best results,

improving significantly over the text-only and video-only methods.[4]

We find that using only YOLO to find visible objects does not provide sufficient information to solve this task. This is due to both the low number of objects that YOLO is able to detect, and the fact that not all actions involve objects. For example, visible actions from our datasets such as "get up", "cut them in half", "getting ready", and "chopped up" cannot be correctly labeled using only object detection. Consequently, we need to use additional video information such as Inception and C3D information.

In general, we find that the text information plays an important role. ELMo embeddings lead to better results than LSTM embeddings, with a relative error rate reduction of 6.8%. This is not surprising given that ELMo uses two bidirectional LSTMs and has improved the state-of-the-art in many NLP tasks [75]. Consequently, we use ELMo in our multimodal model.

Moreover, the addition of extra information improves the results for both modalities. Specifically, the addition of context is found to bring improvements. The use of POS is also found to be generally helpful.

2.7 Conclusion

In this chapter, we address the task of identifying human actions visible in online videos. We focus on the genre of lifestyle vlogs, and construct a new dataset consisting of 1,268 miniclips and 14,769 actions out of which 4,340 have been labeled as visible. We describe and evaluate several text-based and video-based baselines, and introduce a multimodal neural model that leverages visual and linguistic information as well as additional information available in the input data. We show that the multimodal model outperforms the use of one modality at a time.

A distinctive aspect of this work is that we label actions in videos based on the language that accompanies the video. This has the potential to create a large repository of visual depictions of actions, with minimal human intervention, covering a wide spectrum of actions that typically occur in everyday life.

The work described in this chapter has been published in the 57th Annual Meeting of the Association for Computational Linguistics (ACL) [90]. The dataset and the code introduced in this chapter are publicly available at `https://github.com/MichiganNLP/vlog_action_recognition`.

In the next chapter, we explore additional representations and architectures to improve

[4] Significance is measured using a paired t-test: $p < 0.005$ when compared to the best text-only model; $p < 0.0005$ when compared to the best video-only model.

the accuracy of our model, and to identify finer-grained alignments between visual actions and their verbal descriptions.

CHAPTER 3

Human Action Localization

3.1 Introduction

Targeting the long-term goal of video understanding, recent years have witnessed significant progress in the task of action localization, starting with the localization of one action at a time in a short clip [91] or in a longer untrimmed video [92], all the way to localizing more complex natural language queries in videos [14, 93, 94, 95, 96], and recently to localizing complex natural language queries extracted directly from transcripts in online videos [23, 97, 98].

Lifestyle vlogs represent a great challenge and opportunity for this task, as they depict everyday actions in a complex setting. Unlike traditional action datasets [33, 38, 34, 93] or instructional video datasets [97, 23, 22], vlogs contain a wide variety of actions that are more akin to real-life settings, such as "grab my Kindle," "do some reading," or "chill out."

Moreover, vlogs typically include transcripts with complex natural language expressions, which allow us to find an alternative to the costly process of manual annotations. Given the prevalence of vlogs in online platforms, automatically extracting action names from their transcripts can lead to a large-scale inexpensive action dataset. Previous work [22] relied on this technique to build very large datasets of video-action mappings. However, previous work also found that the video and transcript are often misaligned [90, 98]: in the best case, there is a gap of a few seconds between the time when a person verbally expresses the action and when it is visually illustrated.

This chapter addresses the task of temporal action localization in vlogs, and makes three main contributions. First, we introduce a dataset of manual annotations of temporal localization of actions that addresses new challenges compared to other action localization datasets. Second, we present 2SEAL – a simple yet effective method that leverages both language and vision to temporally localize actions, while also accounting for the expected duration of the actions. Through extensive evaluations, we show that our proposed method

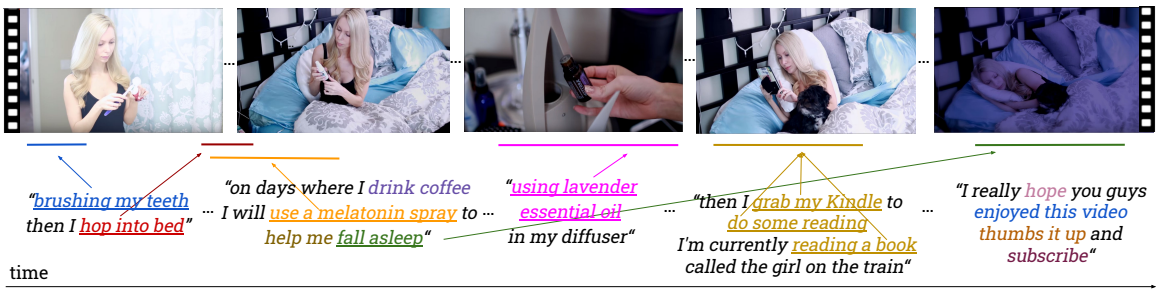


Figure 3.1: Overview of the dataset: distinguishing between actions that are narrated by the vlogger but not visible in the video and actions that are both narrated and visible in the video (underlined), with a highlight on visible actions that represent the same activity (same color). The arrows represent the temporal alignment between when the visible action is narrated as well as the time it occurs in the video. Best viewed in color.

can be used along with existing models to improve their performance on temporal action localization. Finally, we conduct an analysis of the results, and gain insight into the role played by the different components, which further suggests avenues for future work.

3.2 Related Work

Learning connections between vision and language is crucial to many applications. These applications include visual question answering [99, 100, 101], visual content retrieval based on textual queries [22, 102, 103], image and video captioning [63, 104, 100], video summarization with natural language [105, 106], action detection [11, 92, 107], action temporal localization in videos [108, 14, 96, 109, 95] and mapping text descriptions to image or video content [108, 110, 111, 112, 113].

Action Localization Datasets. Action detection and localization algorithms evolve with the building of complex datasets. From searching YouTube videos, given a set of predefined actions [33, 34, 11], or filming in people's homes who act based on a scenario [38], these datasets capture the complexity of daily life activities. However, because of the high annotation cost, these methods are not scalable. Currently, the latest trend in the vision community is to search for pre-defined tasks on WikiHow and collect their corresponding videos from YouTube [22, 23, 97]. This process is more efficient and guarantees that more relevant actions are depicted in the videos. Another technique for collecting human actions is to perform implicit data gathering [4]: instead of explicitly searching for a pre-defined task, find routine videos that contain a broad range of daily actions.

In our work, we use the data introduced in [90] which identifies if the actions mentioned

in the transcripts are present (visible) in the video. Although we use implicit data gathering as proposed in the past, unlike Fouhey et al. [4], who focus on the visual information (hand and object locations), we focus on routine videos that contain rich audio descriptions of the actions being performed, and we use this transcribed audio to extract actions.

Action Localization Methods. Methods that reason over text and visual information do this by first extracting the textual embeddings [114, 115, 40] and visual features [10, 11] and then linearly mapping them to the same embedding space [116, 93, 14, 95]. This is usually computed using self and cross attention over the textual and visual features. The visual features can be extracted with a convolutional neural net as in [116, 117, 118] or from object bounding boxes [111]. Recent work [119, 120, 121] builds on this approach by combining the attention modules in a large scale Transformer architecture [122]. Their goal is to learn inter-modality and cross-modality relationships that can be used in downstream tasks that require complex reasoning about natural language grounded in visual data [123, 124, 125].

Instructional vs. Routine Videos. Action localization methods are moving from using simple pre-defined action labels [11, 107] to more complex natural language action descriptions [116, 126, 22]. Our goal is also to localize natural language descriptions of actions in videos. An important difference between our task and previous work is that the natural language descriptions come from the people filming the actions.

Research work such as [127, 22] also take advantage directly of the actions extracted from the transcripts, however their videos are instructional videos. Instead of looking at instructional videos, we choose a broader category: routine videos, which can contain instructions, but are more focused on describing the typical day of a person.

Compared to instructional videos, routine videos contain a more diverse set of activities, from waking up in the morning and taking a shower, to working out and making a meal. This diversity of actions in one video translates to many more diverse filming perspectives in the same video, which presents a novel challenge for action localization models. Another difference is that routine videos contain higher-level actions that can be abstract in nature (e.g., "wind down," "go for a walk") and thus harder to ground than clear instructions. This is an important difference, as it presents a challenge that is essential for webly supervised systems, which are expected to learn from a diverse mix of both concrete actions and high-level abstract actions. In the realm of web videos, instructional videos account for only a small fraction.

Finally, note that existing action localization methods by and large rely on simplifying assumptions (e.g., instructional videos, always visible actions, non-overlapping actions).

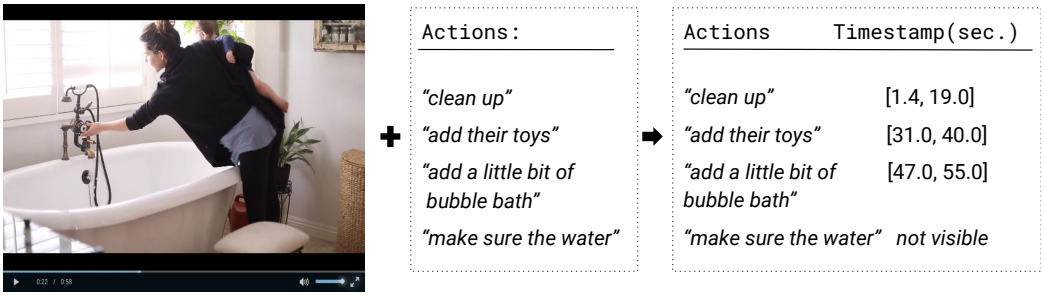


Figure 3.2: Action temporal localization annotation. Each action is localized in the video according to its start and end time offsets. The action is localized according to its visibility in the video, and if it cannot be seen, it is marked as *not visible*.

In contrast, our work introduces an evaluation that accounts for the additional challenges encountered in online videos.

3.3 Data Collection and Annotation

We focus our work on lifestyle vlogs (also known as routine videos), as they are a widely used form of visual information sharing, with tens of millions having been uploaded on YouTube. Vlogs are particularly useful for our research, as they include detailed verbal descriptions of actions. We annotate with temporal information a dataset of vlogs compiled in previous work [90]. Examples from the dataset are shown in Fig. 3.1. We use this particular dataset because the transcripts correlate with the narrations and the actions narrated (mentioned) in the transcripts are already labeled for their property of being visible or not. Although our work focuses mainly on the visible actions, we later use the non visible ones as negative examples.

3.3.1 Vlog Dataset

The dataset that we use as a starting point [90] contains 1,246 video clips from 171 videos drawn from ten different vlog channels. The channels are chosen for their rich content, where the vloggers describe in great detail their routine activities both verbally and visually. Ignat et al. [90] chooses to collect routine videos after observing that by searching explicitly for their name, common everyday actions are hard to find on YouTube. However, collecting routine videos by typing: *my morning routine* or *my after work routine* they discover that this results in millions of videos with human actions (see Table 2.2). By collecting routine videos, instead of searching explicitly for actions, they do implicit data gathering, a form of data collection introduced by [4].

	#actions	Vis. (%)	#videos	#clips
Train	4,939	35.1	110	680
Val	1,264	35.9	26	187
Test	3,456	25.7	35	275

Table 3.1: Statistics for the experimental data split. "Vis." is the percentage of visible actions among the narrated actions.

3.3.2 Temporal Action Annotation

Each video is associated with a set of human actions, in the form of verb phrases extracted from the automatically generated video transcripts. The actions are labeled into two categories: *visible* or *not visible*, depending on whether the actions are explicitly represented in the video. For example, in the video sequence shown in Fig. 3.1, the action "drink coffee" is *not visible* in the video; it is only mentioned as a reason for performing the *visible action* of "use a melatonin spray." Other *not visible* actions from Fig. 3.1 are: "help," "hope," "enjoyed this video," "thumbs it up" and "subscribe," which relate to video feedback but are not visually shown.

Two of the authors of this work annotated the start and end time of all the *visible* actions in the dataset, as illustrated in Fig. 3.2. Each action is localized according to its start and end time offsets. The timestamp is marked according to when the action is visible, which does not necessarily correspond to when it is talked about. If the annotators were not able to localize the action in the clips, they marked it as *not visible*, which corresponds to a correction of the original dataset [90]. They performed the annotations using a simple annotation tool that we built for this purpose, which is publicly available at https://github.com/OanaIgnat/video_annotations.

We measure the inter-annotator agreement by computing the Krippendorff's Alpha score [128] using the interval difference function for each video. We obtain scores between 0.78 and 0.90, which indicate a high agreement.

For our experiments, we split the data by vlog channel. Out of ten channels, six channels are used for training, two channels for validation, and two for testing. Statistics for this experimental split are shown in Table 3.1.

3.3.3 Data Analysis

We perform two types of analyses to gain a better understanding of our dataset.

Action Duration. First, we measure the distribution of action durations in our dataset. As shown later, this information is important, as the action durations can have an impact on

Duration (s)	#actions
0-5	1,136
5-15	1,200
15-25	475
25-35	157
35-45	72
45-60	99

(a)

Long actions	#actions
use (a whisk)	87
make (oatmeal)	81
clean (my skin)	60
Short actions	**#actions**
add (spice)	362
use (the clamps)	228
put (a lid on top)	179

(b)

Dataset	Long actions (%)
Charades-STA [14]	4.2
CrossTask [97]	16.4
COIN [23]	31.6
Ours	25.5

(c)

Table 3.2: Action duration analysis: (a) Distribution in our dataset; (b) Example of long and short actions, each with a sample object, grouped by verbs and sorted by verb frequency; (c) Percentage of long (>15s) actions in other datasets.

the performance of different models. Table 3.2a shows the action duration distribution in the dataset. A summary of *long* actions found in other datasets is shown in Table 3.2c (we define an action as *long* if it exceeds fifteen seconds). Table 3.2b shows examples of *long* actions, grouped by verb and sorted by frequency.

Temporal Relations between Actions. Second, we analyze the temporal relations between actions mentioned in the transcripts. These actions can be challenging to model as they capture the complexities of real life. While there are several actions that follow each other (as more naturally expected), there are also actions that overlap, are included in one another, or even happen at the same time. From a total of 2,070 number of overlapping actions, 1,573 are included in each other and 269 occur exactly at the same time. Table 3.3 shows examples of such actions. While several action localization datasets have been proposed in the past [23], to the best of our knowledge, this dataset is the only action localization dataset that contains *overlapping* actions, making it challenging and novel. For the purpose of this work, we localize each action independent of other actions, but future work may leverage the relations that exist between actions.

Actions that follow each other	Actions that overlap
"make super quick chicken tacos" ; *"do the dishes"*	*"toss everything together"* ∩ *"chop it up"*
"put them in a bowl" ; *"cover in water"*	*"add fresh herbs"* ∩ *"add chickpeas to a bowl"*
"give a little mix" ; *"add half cup of berries"*	*"scoop out of the processor"* ∩ *"scoop it into a bowl"*
"get a little water on your skin" ; *"rinse it off"*	*"combine our dry ingredients"* ∩ *"give it a mix"*
...	...
Actions that are included in each other	**Actions that occur exactly at the same time**
"use a plastic scraper" ⊆ *"wipe thoroughly"*	*"write out"* ≡ *"make your bucket list"*
"throw the cushions around" ⊆ *"fix my cushions up"*	*"go to bed"* ≡ *"head to bed"*
"do this scrub vigorously" ⊆ *"clean some ovens"*	*"add good protein"* ≡ *"use one tablespoon of cashew nut butter"*
"do some yoga" ⊆ *"wind down"*	*"grab my Kindle"* ≡ *"do some reading"*
...	...

Table 3.3: Examples of different types of action temporal relations: actions that overlap (∩), actions that are included in each other (⊆), actions that occur exactly at the same time (≡). From a total of 2,070 number of overlapping actions, 1,573 are included in each other and 269 occur exactly at the same time.

3.4 Two-Stage Action Localization

For a given action mentioned in a video transcript, our goal is to: (1) decide if it is visible within the video clip; and (2) if it is visible, identify its temporal location (i.e., the time interval start and end times).

To achieve this goal, we propose a two-stage method which we call 2SEAL (2-StagE Action Localization).

Figure 3.3 shows the overall architecture of 2SEAL. Following our analysis of the variation in action duration (see Section 3.3.3), and empirical observations made on the development dataset, we hypothesize that shorter actions can be localized mainly based on the temporal information inferred from the transcript (i.e., *when* an action was narrated within the transcript), whereas longer actions are often temporally shifted with respect to their mention in the transcript and thus can benefit from a multimodal model. We thus devise an architecture that first aims to predict whether the action is short or long, and correspondingly activates a transcript alignment (for short actions) or a multimodal model (for long actions). We describe below each of these main components.

Action Duration Classification We use the annotated temporal locations in the videos to determine the expected duration of each action, and build a binary classifier to discriminate between short (≤15s) and long (>15s) actions. We choose this threshold based on the validation data. The classifier uses as input an action text embedding obtained from a text encoder, as described in Section 3.5.

Transcript Alignment. Each video contains a transcript automatically generated by the

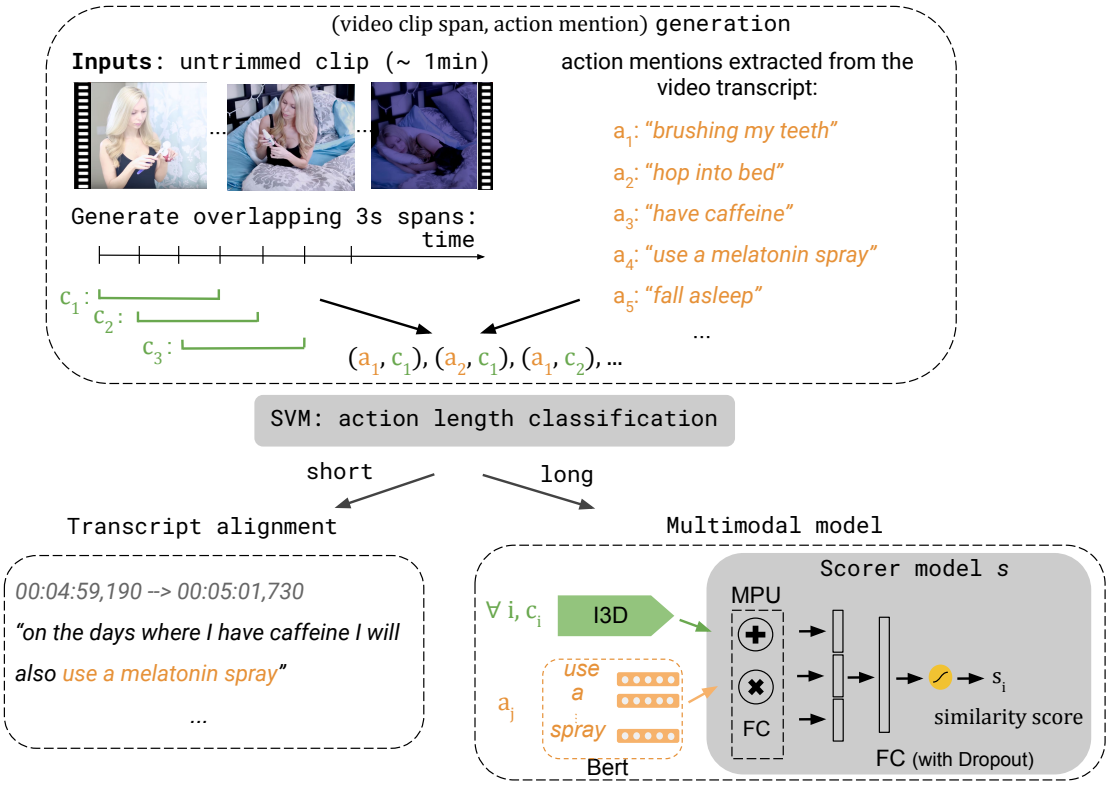


Figure 3.3: 2SEAL method architecture. Note the depicted MPU-based multimodal model can be replaced with any multimodal model. The MPU model is composed of vector element-wise addition ('+'), vector element-wise multiplication ('x') and vector concatenation followed by a Fully Connected ('FC') layer to combine the information from both textual and visual modalities.

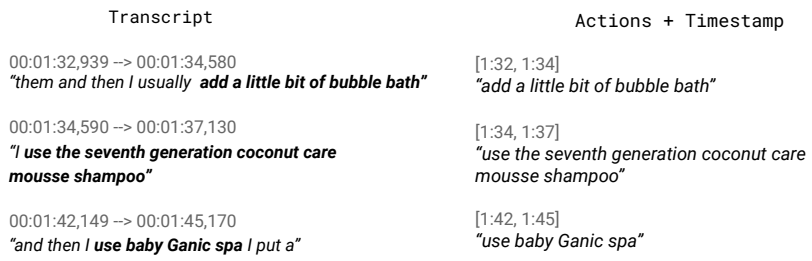


Figure 3.4: Example of applying the Transcript Alignment method. The transcript contains time intervals for utterances. Each action contained in an utterance is assigned the corresponding time interval.

YouTube API. The transcript contains time information for every utterance. Given an action mention extracted from an utterance, the Transcript Alignment method assumes the action is visible, and predicts its temporal location to be the time interval associated with the corresponding utterance, as illustrated in Fig. 3.4. The transcript alignment is also illustrated in Fig. 3.3.

Multimodal Model. We split the video clips into fixed-duration spans and convert the action temporal localization task into binary classification tasks based on the output from a scorer model s. We aim to predict if the visual information from a video clip span corresponds to the linguistic representation of an action. For a given action mention within the transcript and a fixed-duration video clip span, we compute a similarity score to decide if they correspond to each other. The action mention is represented using a text encoder and the features for the video clip span are obtained from a video encoder (see Section 3.5).

The process of pairing action mentions to video clip spans is depicted in Fig. 3.3. s can be represented by any multimodal model, and we describe several models in Section 3.5.2. At test time, given a video clip and its corresponding transcript, we input all the pairs of action mentions and fixed-duration video clip spans. We merge all the spans that surpass a certain threshold and are separated by less than three seconds into *proposals*. Each proposal is assigned the maximum similarity score of its spans. We then perform non-maximum suppression to select the best proposal as the predicted action location interval. At training time, we focus only on the binary task and train s with the standard cross-entropy loss. Given that an action mention has many more negative (*not visible*) fixed-duration video clip spans in a given video clip, we balance the classes out via downsampling by taking negative random samples from the same video clip. The question of how different negative sampling strategies affect the scorer model performance is left for future work.

Method	A	P	R	F1
Majority	74.4	74.4	100.0	85.3
Action Duration Clf.	80.6	81.8	97.6	89.0

Table 3.4: Action duration classification results on the validation set. The classification is binary, where the positives are the short actions ($\leq$15s) and the negatives the long ones ($>$15s). The columns are in order: accuracy (A), precision (P), recall (R) and F1 score (F1).

3.5 Experiments

To evaluate our duration-informed action localization method, we run several comparative experiments on the dataset described in Section 3.3. We compare our method with several strong baselines, and also perform feature ablation and a breakdown of results by action duration.

In all our experiments, we use a video encoder consisting of the last layer (`mixed_5c`) from a Kinetics [11] pre-trained I3D model. The video clips are divided into overlapping three-second spans with a stride of 1s. We freeze both the text and the video encoders and take their outputs as features. For the Action Duration Classification, we use an SVM classifier with C=1.0 and an RBF kernel, and weight the samples inversely proportional to their class frequency. We train the models using an Adam optimizer with early stopping (tolerance 15 epochs), with a learning rate of 0.001 and a batch size of 64.

3.5.1 Action Duration Classification

We train the action duration classifier described in the previous section using only the visible actions. The results are reported in Table 3.4. For comparison, we also show the performance of a majority classifier, which labels every action as "short" by default. As shown in the table, despite the simplicity of the classifier, the action duration classifier obtains good improvement over the majority baseline.

3.5.2 Temporal Action Localization

Our 2SEAL method includes a scorer that measures the similarity between a video clip and an action mention (see Fig. 3.3). To implement this scorer, we experiment with three methods proposed in previous work: multimodal processing unit, multiple instance learning noise contrastive estimation, and stacked cross attention.

Multimodal Processing Unit (MPU). We use the MPU model [14] to compute the similar-

ity score between the language representation of a narrated action and a video clip span. For the text features, we fine-tune a pre-trained BERT-base-uncased [40] for domain adaption by on 884 vlog transcripts with 80,749 sentences. We take embeddings from this model for the action mentions in the transcripts by average pooling (the final embedding size is 768). In Section 3.6.2 we experiment with variations of this text encoder. The text and visual features for each pair are linearly mapped to the same embedding space. Next, the MPU model is applied to compute the interaction between the two vectors of the same duration. The MPU model is composed of vector element-wise addition ('+'), vector element-wise multiplication ('x') and vector concatenation followed by a Fully Connected ('FC') layer to combine the information from both textual and visual modalities. The outputs from all three operations are concatenated to construct a multimodal representation. This process is also illustrated in the overall architecture in Fig. 3.3. The resulting representation is given as input to a linear layer and finally to a sigmoid function to obtain a similarity score.

Multiple Instance Learning Noise Contrastive Estimation (MIL-NCE). We use the MIL-NCE model from [98] which was trained on HowTo100M [22]. The similarity score is computed as a dot product between the text and video encoder outputs. The text encoder takes embeddings from a GoogleNews-pretrained skipgram word2vec [129] implementation and further processes and pools the embeddings to obtain a fixed-size representation. We use the MIL-NCE I3D[1] visual features, and not the S3D features, for consistency reasons and to ensure a fair comparison between the multimodal models. We empirically find it beneficial to threshold the similarities at mid-range value after experimenting with linear regression models on the validation data. Note we do not fine-tune this model but freeze it. Future work can explore how the method benefits from fine-tuning.

Stacked Cross Attention (SCA). We also experiment with the SCA method [111], and adapt its *Text-Image* formulation. It first attends to image frames with respect to each word, and then compares each word to its corresponding attended frame vector to determine the importance of each word. The relevance R between the i-th word and the image is defined as the cosine similarity between the i-th word vector v_i and its attended frame vector a_i^t. The final similarity score between image I and sentence T is summarized by average pooling: $S'_{AVG}(I,T) = \frac{1}{n}\sum_{j=1}^{n} R'(e_j, a_j^v)$. The textual features are represented using a Gated Recurrent Unit (GRU) [130] as in [111]. We use the mid-range threshold for the similarity score.

2D Temporal Adjacent Networks (2D-TAN). We find the 2D-TAN model [15] suitable for our task as it is built to localize multiple natural language queries in a video.

[1] https://tfhub.dev/deepmind/mil-nce/i3d/1

The video clips are represented using C3D [64] features and the action queries using GloVe [114] embeddings, as described in the 2D-TAN paper [15]. We take as final proposal the action localization proposal with the highest score.

We test the pre-trained model and also fine-tune it on our training and validation data. We run two model configurations, which were trained on TACoS [108], namely "Pool" and "Conv" in our test set. "Pool" and "Conv" represent max-pooling and stacked convolution respectively, which indicates two different ways for moment feature extraction in the 2D-TAN model. We report the results of fine-tuned "Conv" 2D-TAN model, which is the best performing 2D-TAN model configuration on our test dataset.

3.5.3 Results

We evaluate the predictions made by the action localization methods using two evaluation metrics. First, we compute the Visibility Accuracy (VA) to decide if the method can distinguish between visible and not visible actions. Second, only for the visible actions, we compute the recall at different Intersection over Union (IoU) thresholds: 0.1, 0.3, 0.5 and 0.7. A higher threshold means a stronger constraint on how exact the match between the predicted and the ground truth location needs to be. If the predicted interval has an IoU score with the ground truth greater than the threshold, we consider the prediction as being correct. We also compute the average recall over all IoU values, as the mIoU. Note that if a method predicted that a visible action is non-visible, then the recall score is penalized.

Table 3.5 presents the temporal action localization results on our data. The Transcript Alignment method performs better than the MPU, MIL-NCE, SCA and 2D-TAN methods if we do not previously apply our proposed 2SEAL method before. However, when using our 2SEAL method that combines both the Transcript Alignment and a method to score long actions (either MPU, MIL-NCE, SCA, or 2D-TAN), the performance improves significantly, with the system integrating the MPU model leading to the best results. We suspect MIL-NCE may perform better if fine-tuned, however our intention is not to compare MPU and MIL-NCE but to show how our method can improve over other existing methods. The results confirm our initial hypothesis that actions of different duration benefit from different methods: the transcript alignment excels at *short* actions, while the multimodal model performs better for *long* actions.

Method	VA	Recall IoU=0.1	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
All visible	25.7	67.4	23.6	8.3	4.1	21.6
All non-visible	74.3	0.0	0.0	0.0	0.0	0.0
Transcript Alignment (ours)	25.7	73.3	47.3	22.2	7.2	30.8
MPU	75.5	57.9	27.0	12.4	6.2	21.4
2SEAL (ours) + MPU	**79.0**	**74.6**	**48.7**	**22.8**	**8.6**	**31.9**
MIL-NCE	26.1	62.9	22.2	8.0	4.2	20.5
2SEAL (ours) + MIL-NCE	34.4	74.4	47.8	21.7	7.9	31.4
SCA	24.2	49.9	17.0	6.0	3.4	15.9
2SEAL (ours) + SCA	26.1	72.2	46.7	21.4	7.6	30.5
2D-TAN	25.7	49.4	23.1	10.9	3.7	17.6
2SEAL (ours) + 2D-TAN	25.7	73.4	47.0	21.6	7.7	30.8
Human	85.9	83.5	71.8	52.0	35.0	50.3

Table 3.5: Results on the test set. "VA" stands for Visibility Accuracy.

3.6 Analyses and Discussion

To gain insights into the performance of our proposed model in relation to action duration, and to understand the role played by different features, we perform several analyses.

3.6.1 Action Duration Impact

If the action is brief, the IoU metric will be influenced by a few seconds compared to when the action is longer in duration. This metric penalizes more the mislocalization of short actions, as compared to the longer ones. This analysis is often done for the task of object detection, where the IoU scores are grouped by bounding box size [131]. To verify our initial hypothesis that actions of different duration benefit from different localization methods, we break down the results of the MPU (the best scorer from among MPU, MIL-NCE, SCA and 2D-TAN without applying the 2SEAL method) by action duration in Table 3.6. As shown in the table, the performance of the model is connected to the duration of the actions. For *long* actions, the multimodal method obtains better results compared to the transcript alignment method, while the opposite is true for *short* actions.

	0–15s		16–35s		36–60s	
Recall	MPU	Align	MPU	Align	MPU	Align
IoU=0.1	49.5	**71.6**	**90.7**	76.6	**95.2**	83.3
IoU=0.3	5.4	**49.0**	**73.4**	51.4	**81.0**	0.0
IoU=0.5	2.0	**25.0**	**22.0**	17.8	**78.6**	0.0
IoU=0.7	0.8	**9.4**	**5.6**	1.9	**66.7**	0.0
mIoU	12.0	**32.0**	**38.9**	29.9	**71.7**	16.5

Table 3.6: Breakdown by action duration (time span) on the validation set. The MPU model performance increases with the increase of action time span, while transcript alignment (Align) performance decreases.

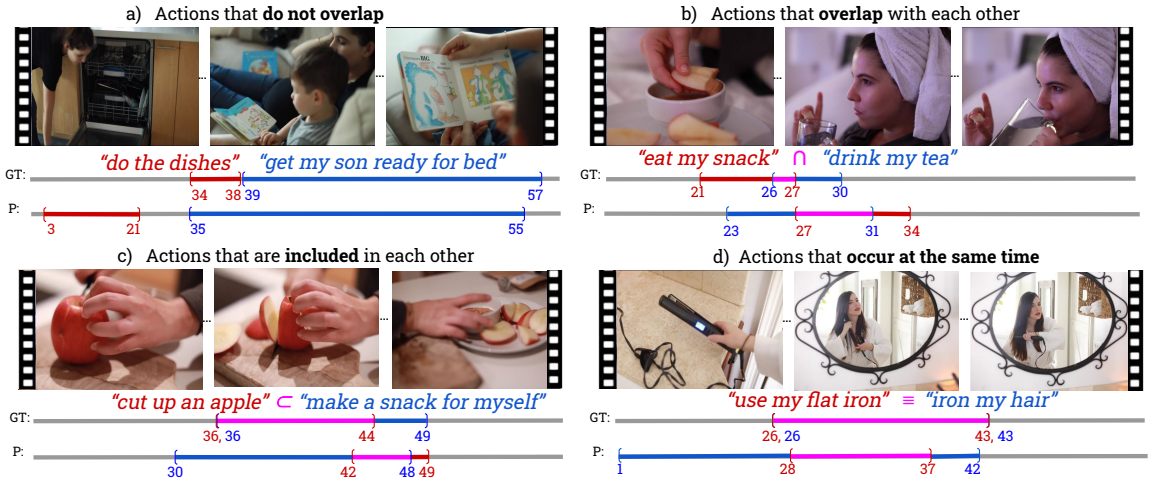


Figure 3.5: Randomly sampled qualitative results for different cases of action overlapping. Best viewed in color.

Method	Recall IoU=0.1	IoU=0.3	IoU=0.5	IoU=0.7	mIoU
MPU	**57.9**	**27.0**	**12.4**	**6.2**	**21.4**
MPU verb only	33.5	18.5	9.2	4.8	13.7
MPU verb+noun only	33.8	18.7	9.8	4.8	14.0
MPU BERT w/o DA	46.9	26.4	14.1	5.4	19.0
MPU ELMo	48.5	23.7	10.6	6.2	18.4
MPU GloVe	41.6	22.5	11.6	6.9	17.2
MPU video only	41.5	25.4	13.9	6.8	18.0
MPU text only	25.3	11.6	4.3	2.2	9.1

Table 3.7: Results on the test set for different variations of the input to the MPU model. "DA" stands for Domain Adaptation.

3.6.2 Text and Visual Features

In Table 3.7, we experiment with the MPU model (without applying the 2SEAL method) and look into how each modality contributes to solving this task, by removing one modality at a time from our best performing model. We also analyze other types of text embeddings. Inspired by [132, 133], we focus on verbs and nouns, which we extract from the actions and compute their BERT embeddings. We observe that the visual information contributes the most to the task of action localization, as removing this information drastically lowers the model performance. Another observation is that processing the entire action is more beneficial to the model than focusing only on nouns and verbs.

3.6.3 Qualitative Results

Randomly sampled results are shown in Fig. 3.5. They are grouped by the different levels of action overlapping: no overlap, intersection, inclusion and perfect overlap. From analyzing these results, a future work direction emerges: detecting which actions are likely to happen at the same time can lead to better algorithms for action localization.

3.7 Conclusion

In this chapter, we introduced a new dataset for action localization in vlogs — a growing form of online video communication where everyday routine actions are described in language and also presented visually. Using this dataset, we addressed the task of temporal

action localization in videos. We proposed 2SEAL – a simple yet effective method to visually localize the actions mentioned in a video transcript, which relies on both language and vision, and specifically accounts for the duration of an action for the purpose of building a more accurate system.

Through several extensive evaluations, we showed that our method improves and complements other methods by first computing the expected duration of an action, and selectively applying a language-based or multimodal model depending on the action duration. This work contributes to the larger body of work for multimodal understanding, and at the same time builds a large repository of vision-language representations covering a wide spectrum of actions that can be used for downstream tasks such as action recognition systems, human behavior understanding, event recognition, and others.

The work described in this chapter has been accepted with minor revisions in ACM Transactions on Multimedia Computing, Communications, and Applications. The dataset introduced in this chapter, the annotation tool, and the system code are publicly available at https://github.com/MichiganNLP/vlog_action_localization.

In the next chapter we introduce how we can use multimodal information from lifestyle vlogs for developing models for human behavior understanding.

CHAPTER 4

Human Action Reason Identification

4.1 Introduction

Significant research effort has been recently devoted to the task of action recognition [11, 134, 135, 136, 137, 138]. Action recognition works well when applied to well defined/constrained scenarios, such as people following scripts and instructions [38, 22, 23], performing sports [16, 139] or cooking [18, 20, 140, 21]. At the same time however, action recognition is limited and error-prone once the application space is opened to everyday life. This indicates that current action recognition systems rely mostly on pattern memorization and do not effectively understand the action, which makes them fragile and unable to adapt to new settings [141, 142]. Research on how to improve action recognition in videos [141] shows that recognition systems for actions with known intent have a significant increase in performance, as knowing the reason for performing an action is an important step for understanding that action [143, 144].

In contrast to action recognition, action causal reasoning research is just emerging in computational applications [145, 146, 147, 148]. Causal reasoning has direct applications on many real-life settings, for instance to understand the consequences of events (e.g., if "there is clutter," "cleaning" is required), or to enable social reasoning (e.g., when "guests are expected," "cleaning" may be needed – see Figure 4.1). Most of the work to date on causal systems has relied on the use of semantic parsers to identify reasons [149], however this approach does not work well on more realistic every-day settings As an example, consider the statement "This is a mess and my friends are coming over. I need to start cleaning." Current causal systems are unable to identify "this is a mess" and "friends are coming over" as reasons, and are thus failing to use them as context for understanding the action of "cleaning."

In this chapter, we propose the task of multimodal action reason identification in everyday life scenarios. We collect a dataset of lifestyle vlogs from YouTube that reflect

Figure 4.1: Overview of our task: automatic identification of action reasons in online videos. The reasons for *cleaning* change based on the visual and textual (video transcript) context. The videos are selected from YouTube, and the actions together with their reasons are obtained from the ConceptNet knowledge graph which we supplement with crowd-sourced reasons. The figure shows two examples from our WHYACT dataset.

daily scenarios and are currently very challenging for systems to solve. Vloggers freely express themselves while performing most common everyday activities such as cleaning, eating, cooking, writing and others. Lifestyle vlogs present a person's everyday routine: the vlogger visually records the activities they perform during a normal day and verbally express their intentions and feelings about those activities. Because of these characteristics, lifestyle vlogs are a rich data source for an in depth study of human actions and the reasons behind them.

The work makes four main contributions. First, we formalize the new task of multimodal action reason identification in online vlogs. Second, we introduce a new dataset, WHYACT, consisting of 1,077 (action, context, reasons) tuples manually labeled in online vlogs, covering 24 actions and their reasons drawn from ConceptNet as well as crowd-sourcing contributions. Third, we propose several models to solve the task of human action reason identification, consisting of single-modalities models based on the visual content and vlog transcripts, as well as a multimodal model using a fill-in-the-blanks strategy. Finally, we also present an analysis of our new dataset, which leads to rich avenues for future work for improving the tasks of reason identification and ultimately action recognition in online videos.

4.2 Related Work

There are three areas of research related to our work: identifying action motivation, commonsense knowledge acquisition, and web supervision.

Identifying Action Motivation. The research most closely related to our work is the work that introduced the task of predicting motivations of actions by leveraging text [145]. Their method was applied to images from the COCO dataset [150], while ours is focused on videos from YouTube. Other work on human action causality in the visual domain [146, 147] relies on object detection and automatic image captioning as a way to represent videos and analyze visual causal relations. Research has also been carried out on detecting the intentions of human actions [151]; the task definition differs from ours, however, as their goal is to automatically choose the correct action for a given image and intention. Other related work includes [152], a vision-based classification model between intentional and non-intentional actions and Intentonomy [153], a dataset on human intent behind images on Instagram.

Commonsense Knowledge Acquisition. Research on commonsense knowledge often relies on textual knowledge bases such as ConceptNet [6], ATOMIC [7], COMET-ATOMIC 2020 [28], and more recently GLUCOSE [154].

Recently, several of these textual knowledge bases have also been used for visual applications, to create more complex multimodal datasets and models [155, 148, 156]. VisualCOMET [155] is a dataset for visual commonsense reasoning tasks to predict events that might have happened before a given event, events that might happen next, as well as people intents at a given point in time. Their dataset is built on top of VCR [157], which consists of images of multiple people and activities. Video2Commonsense [148] uses ATOMIC to extract from an input video a list of intentions that are provided as input to a system that generates video captions, as well as three types of commonsense descriptions (intention, effect, attribute). KVL-BERT [156] proposes a knowledge enhanced cross-modal BERT model by introducing entities extracted from ConceptNet [6] into the input sentences, followed by testing their visual question answering model on the VCR benchmark [157]. Unlike previous work that broadly addresses commonsense relations, we focus on the extraction and analysis of action reasons, which allows us to gain deeper insights for this relation type.

Webly-Supervised Learning. The space of current commonsense inference systems is often limited to one dataset at a time, e.g., COCO [150], VCR [157], MSR-VTT [158].

In our work, we ask commonsense questions in the context of rich, unlimited, constantly evolving online videos from YouTube.

Previous work has leveraged webly-labeled data for the purpose of identifying commonsense knowledge. One of the most extensive efforts is NELL (Never Ending Language Learner) [159], a system that learns everyday knowledge by crawling the web, reading documents and analysing their linguistic patterns. A closely related effort is NEIL (Never Ending Image Learner), which learns commonsense knowledge from images on the web [160]. Large scale video datasets [22] on instructional videos and lifestyle vlogs [4, 90] are other examples of web supervision. The latter are similar to our work as they analyse online vlogs, but unlike our work, their focus is on action detection and not on the reasons behind actions.

4.3 Data Collection and Annotation

In order to develop and test models for recognizing reasons for human actions in videos, we need a manually annotated dataset. This section describes the WHYACT dataset of action reasons.

4.3.1 Data Collection

We start by compiling a set of lifestyle videos from YouTube, consisting of people performing their daily routine activities, such as cleaning, cooking, studying, relaxing, and others. We build a data gathering pipeline to automatically extract and filter videos and their transcripts.

We select five YouTube channels and download all the videos and their transcripts. The channels are selected to have good quality videos with automatically generated transcripts containing detailed verbal descriptions of the actions depicted. An analysis of the videos indicates that both the textual and visual information are rich sources for describing not only the actions, but why the actions in the videos are undertaken (action reasons). We present qualitative and quantitative analyses of our data in section 4.6.

We also collect a set of human actions and their reasons from ConceptNet [6]. Actions include verbs such as: *clean*, *write*, *eat*, and other verbs describing everyday activities. The actions are selected based on how many reasons are provided in ConceptNet and how likely they are to appear in our collected videos. For example, the action of *cleaning* is likely to appear in the vlog data, while the action of *yawning* is not.

Initial	9,759
Actions with reasons in ConceptNet	139
Actions with at least 3 reasons in CN	102
Actions with at least 25 video-clips	25

Table 4.1: Statistics for number of collected actions at each stage of data filtering.

4.3.2 Data Pre-processing

After collecting the videos, actions and their corresponding reasons, the following data pre-processing steps are applied.

Action and Reason Filtering. From ConceptNet, we select actions that contain at least three reasons. The reasons in ConceptNet are marked by the "motivated by" relation. We further filter out those actions that appear less than 25 times in our video dataset, in order to assure that each action has a significant number of instances.

We find that the reasons from ConceptNet are often very similar to each other, and thus easy to confound. For example, the reasons for the action *clean* are: "dirty", "remove dirt", "don't like dirtiness", "there dust", "dirtiness unpleasant", "dirt can make ill", "things cleaner", "messy", "company was coming". To address this issue, we apply agglomerative clustering [161] to group similar actions together. For instance, for the action *clean*, the following clusters are produced: ["dirty", "remove dirt", "there dust", "things cleaner"], ["don't like dirtiness", "dirtiness unpleasant", "dirt can make ill"], ["messy"], ["company was coming"]. Next, we manually select the most representative and clear reason from each cluster. We also correct any spelling mistakes and rename the reasons that are either too general or unclear (e.g., we rename "messy" to "declutter"). Finally, after the clustering and processing steps, we filter out all the actions that contain less than three reasons.

We show the statistics before and after the additive filtering steps in Table 4.1.

Transcript Filtering. We want transcripts that reflect the reasons for performing one or more actions shown in the video. However, the majority of the transcripts contain mainly verbal descriptions of the action, which are not always helpful in determining their reason. We therefore implement a method to select candidate transcript sequences that contain at least one causal relation related to the actions shown in the video.

We start by automatically splitting the transcripts into sentences using spaCy [162]. Next, we select the sentences with at least one action from the final list of actions we collected from ConceptNet (see the previous section). For each selected sentence, we also

collect its context consisting of the sentences before and after. We do this in order to increase the search space for the reasons for the actions mentioned in the selected sentences.

We want to keep the sentences that contain action reasons. We tried multiple methods to automatically determine the sentences more likely to include causal relations using Semantic Role Labeling (SRL) [163], Open Information Extraction (OpenIE) [164] and searching for causal markers. We found that SRL and OpenIE do not work well on our data, likely due to the fact that the transcripts are more noisy than the datasets these models were trained on. Most of the language in the transcripts does not follow simple patterns such as "I clean because it is dirty." Instead, the language consists of natural everyday speech such as "Look at how dirty this is, I think I should clean it."

We find that a strategy sufficient for our purposes is to search for causal markers such as "because", "since", "so that is why", "thus", "therefore" in the sentence and the context, and constrain the distance between the actions and the markers to be less than 15 words – a threshold identified on development data. We thus keep all the transcript sentences and their context that contain at least one action and a causal marker within a distance of less than the threshold of 15 words.

Video Filtering. As transcripts are temporally aligned with videos, we can obtain meaningful video clips related to the narration. We extract video clips corresponding to the sentences selected from transcripts (described in the section above).

We want video clips that show why the actions are being performed. Although there can be many actions along with reasons in the transcript, if they are not depicted in the video, we cannot leverage the video information in our task. Videos with low movement tend to show people sitting in front of the camera, describing their routine, but not performing the action they are talking about. We therefore remove clips that do not contain enough movement. We sample one out of every one hundred frames of the clip, and compute the 2D correlation coefficient between these sampled frames. If the median of the obtained values is greater than a certain threshold (0.8, selected on the development data), we filter out the clip. We also remove video-clips that are shorter than 10 seconds and longer than 3 minutes.

4.3.3 Data Annotation

The resulting (video clip, action, reasons) tuples are annotated with the help of Amazon Mechanical Turk (AMT) workers. They are asked to identify: (1) what are the reasons shown or mentioned in the video clip for performing a given action; (2) how are the reasons

Please carefully read the instructions before performing the task.

▼ Instructions

You are given a video that contains a person describing an action and a list of **candidate reasons for why they want to do the action.**

From the list of candidate **reasons**, select the ones that are mentioned verbally or shown visually in the video.

What are the reasons shown or mentioned in the video for performing the action of cleaning?

Please select one or more categories:

- ☐ company was coming
- ☐ do not like dirtiness
- ☐ declutter
- ☐ remove dirt
- ☐ I cannot find any reason mentioned verbally or shown visually in the video

Please select how did you find the reasons in the video:

- ○ The reasons are mentioned verbally
- ○ The reasons are shown visually
- ○ The reasons are mentioned verbally and shown visually

Please select how confident are you in your answers:

- ○ High confidence
- ○ Low confidence

If there are other reasons that you found, please write them here.

Figure 4.2: Instructions for the annotators.

Video-clips	1,077
Video hours	107.3
Transcript words	109,711
Actions	24
Reasons	166

Table 4.2: Data statistics.

	Test	Development
Actions	24	24
Reasons	166	166
Video-clips	853	224

Table 4.3: Statistics for the experimental data split. The methods we run are unsupervised with fine-tuning on development set.

identified in the video: are they mentioned verbally, shown visually, or both; (3) whether there are other reasons other than the ones provided; (4) how confident the annotator is in their response. The guidelines and interface for annotations are shown in Figure 4.2. In addition to the guidelines, we also provide the annotators with a series of examples of completed assignments with explanations for why the answers were selected. We present them in Figure 4.3.

We add new action reasons from the ones added by the annotators if they repeat at least three times in the collected answers and are not similar to the ones already existing.

Each assignment is completed by three different master annotators. We compute the agreement between the annotators using Fleiss Kappa [165] and we obtain 0.6, which indicates a moderate agreement. Because the annotators can select multiple reasons, the agreement is computed per reason and then averaged.

We also analyse how confident the workers are in their answers: for each video, we take the confidence selected by the majority of workers: out of 1,077 videos, in 890 videos the majority of workers are highly confident.

Table 4.2 shows statistics for our final dataset of video-clips and actions annotated with their reasons. Figure 4.1 shows a sample video and transcript, with annotations. Additional examples of annotated actions and their reasons can be seen in Figure 4.4.

4.4 Identifying Causal Relations in Vlogs

Given a video, an action, and a list of candidate action reasons, our goal is to determine the reasons mentioned or shown in the video. We develop a multimodal model that leverages

both visual and textual information, and we compare its performance with several single-modality baselines.

The models we develop are unsupervised in that we are not learning any task-specific information from a training dataset. We use a validation set only to tune the hyper-parameters of the models.

4.4.1 Data Processing and Representation

Textual Representations. To represent the textual data – transcripts and candidate reasons – we use sentence embeddings computed using the pre-trained model Sentence-BERT [166].

Video Representations. In order to tie together the causal relations, both the textual, and the visual information, we represent the video as a bag of object labels and a collection of video captions. For object detection we use Detectron2 [167], a state-of-the-art object detection algorithm.

We generate automatic captions for the videos using a state-of-the-art dense captioning model [168]. The input to the model are visual features extracted from I3D model pre-trained on Kinetics [11], audio features extracted with VGGish model [169] pre-trained on YouTube-8M [34] and caption tokens using GloVe [114].

4.4.2 Baselines

Using the representations described in Section 4.4.1, we implement several textual and visual models.

4.4.2.1 Textual Similarity

Given an action, a video transcript associated with the action, and a list of the candidate action reasons, we compute the cosine similarity between the textual representations of the transcript and all the candidate reasons. We predict as correct those reasons that have a cosine similarity with the transcript greater than a threshold of 0.1. The threshold is fine-tuned on development data.

Because the transcript might contain information that may be unrelated to the action described or its reasons, we also develop a second version of this baseline. When computing the similarity, instead of using the whole transcript, we select only the part of the transcript that is in the vicinity of the causal markers (before and after a fixed number words,

fine-tuned on development data).

4.4.2.2 Natural Language Inference (NLI)

We use a pre-trained NLI model [170] as a zero-shot sequence classifier. The NLI model is pre-trained on the Multi-Genre Natural Language Inference (MultiNLI) corpus [171], a collection of sentence pairs annotated with textual entailment information.

The method works by posing the sequence to be classified as the NLI premise and constructing a hypothesis from each candidate label: given the transcript as a premise and the list of reasons as the hypotheses, each reason will receive a score that reflects the probability of entailment. For example, if we want to evaluate whether the label "declutter" is a reason for the action "cleaning", we construct the hypothesis "The reason for cleaning is declutter."

We use a threshold of 0.8 fine-tuned on the development data to filter the reasons that have a high entailment score with the transcript.

Bag of Objects. We replace the transcript in the premise with a list of object labels detected from the video. The objects are detected using the Detectron2 model [167] on each video frame, at 1fps. We select only the objects that pass a confidence score of 0.7.

Automatic Video Captioning. We replace the transcript in the premise with a list of video captions detected using the Bi-modal Transformer for Dense Video Captioning model [168]. The video captioning model generates captions for several time slots. We further filter the generated captions to remove redundant captions: if a time slot is heavily overlapped or even covered by another time slot, we only keep the caption of the longer time slot. We find that captions of longer time slots are also more informative and accurate compared to captions of shorter time slots.

4.4.3 Multimodal Model

To leverage information from both the visual and linguistic modalities, we propose a new model that recasts our task as a Cloze task, and attempts to identify the action reasons by performing a fill-in-the-blanks prediction, similarly to Castro et al. [172] that proposes to fill blanks corresponding to noun phrases in descriptions based on video clips content. Specifically, after each action mention for which we want to identify the reason, we add the text "because _____." For instance, "I clean the windows" is replaced by "I clean the windows because _____". We train a language model to compute the likelihood of filling

in the blank with each of the candidate reasons. For this purpose, we use T5 [42], an encoder-decoder transformer [173] pre-trained model, to fill in blanks with text.

To incorporate the visual data, we first obtain Kinetics-pre-trained I3D [11] RGB features at 25fps (the average pooling layer). We input the features to the T5 encoder after the transcript text tokens. The text input is passed through an embedding layer (as in T5), while the video features are passed through a linear layer. Since T5 was not trained with this kind of input, we fine-tune it on unlabeled data from the same source, without including data that contains the causal marker "because". Note this also helps the model specialize on filling-in-the-blank with reasons. Finally, we fine-tune the model on the development data. We obtain the reasons for an action by computing the likelihood of the potential ones and taking the ones that pass a threshold selected based on the development data. The model architecture is shown in Figure 4.5.

We also use our fill-in-the-blanks model in a single modality mode, where we apply it only on the transcript.

4.5 Evaluation

We consider as gold standard the labels selected by the majority of workers (at least two out of three workers).

For our experiments, we split the data across video-clips: 20% development and 80% test (see Table 4.3 for a breakdown of actions, reasons and video-clips in each set). We evaluate our systems as follows. For each action and corresponding video-clip, we compute the Accuracy, Precision, Recall and F1 scores between the gold standard and predicted labels. We then compute the average of the scores across actions. Because the annotated data is unbalanced (in average, 2 out of 6 candidate reasons per instance are selected as gold standard), the most representative metric is F1 score. The average results are shown in Table 4.4. The results also vary by action: the F1 scores for each action, of the best performing method, are shown in Section 4.5.

Experiments on WHYACT reveal that both textual and visual modalities contribute to solving the task. The results demonstrate that the task is challenging and there is room for improvement for future work models.

Selecting the most frequent reason for each action on test data achieves on average an F1 of 40.64, with a wide variation ranging from a very low F1 for the action "writing" (7.66 F1) to a high F1 for the action "cleaning" (55.42 F1). Note however that the "most frequent reason" model makes use of data distributions that our models do not use (because our models are not trained). Furthermore, we believe that it is expected that for certain actions

the distribution of reasons is unbalanced, as in everyday life there are action reasons much more common than others (e.g., for "cleaning", "remove dirt" is a more common/frequent reason than "company was coming").

Method	Input	Accuracy	Precision	Recall	F1
	BASELINES				
Cosine similarity	Transcript	57.70	31.39	55.94	37.64
	Causal relations from transcript	50.85	30.40	68.91	39.73
	SINGLE MODALITY MODELS				
Natural Lan-guage Inference	Transcript	**68.41**	**41.90**	48.01	40.78
	Video object labels	54.49	31.70	59.93	36.79
	Video dense captions	49.18	29.54	68.47	37.40
	Video object labels & dense captions	36.93	27.34	87.97	39.11
Fill-in-the-blanks	Transcript	44.04	30.70	87.10	**43.59**
	MULTIMODAL NEURAL MODELS				
Fill-in-the-blanks	Video & Transcript	32.6	27.56	**94.76**	41.11

Table 4.4: Results from our models on test data.

4.6 Data Analysis

We perform an analysis of the actions, reasons and video-clips in the WHYACT dataset. The distribution of actions and their reasons are shown in Figure 4.6, together with additional analyses: the distribution of actions and their number of reasons (Section 4.5) and videos (Section 4.5) and the distribution of actions and their worker agreement scores (Section 4.5).

We also explore the content of the videos by analysing their transcripts. In particular, we look at the actions and their direct objects. For example, the action clean is depicted in various ways in the videos: "clean shower", "clean body", "clean makeup", "clean dishes". The action diversity assures that the task is challenging and complex, trying to cover the full spectrum of everyday activities. In Figure 4.8 we show what kind of actions are depicted in the videos: we extract all the verbs and their most five most frequent direct objects using spaCy [162] and then we cluster them by verb and plot them using t-distributed Stochastic Neighbor Embedding (t-SNE) [174].

Finally, we analyse what kind of information is required for detecting the action reasons: what is verbally described, visually shown in the video or the combination of visual and verbal cues. For this, we analyse the worker's justifications for selecting the action reasons: if the reasons were verbally mentioned in the video, visually shown or both. For each video, we take the justification selected by the majority of workers. We find that the reasons for the actions can be inferred only by relying on the narration for less than half of the videos (496 / 1,077). For the remaining videos, the annotators answered that they relied on either the visual information (in 55 videos) or on both visual and audio information (in 423 videos). The remaining 103 videos do not have a clear agreement among annotators on the modality used to indicate the action reasons. We believe that this imbalanced split might be a reason for why the multimodal model does not perform as well as the text model. For future work, we want to collect more visual data that contains action reasons.

Impact of reason specificity on model performance. The reasons in WHYACT vary from specific (e.g., for the verb "fall', possible reasons are: "tripped", "ladder broke", "rush", "makeup fell") to general (e.g., for the verb "play", possible reasons are: "relax", "entertain yourself", "play an instrument"). We believe that a model can benefit from learning both general and specific reasons. From general reasons such as "relax", a model can learn to extrapolate, generalize, and adapt to other actions for which those reasons might apply (e.g., "relax" can also be a reason for actions like "drink" or "read") and use these general reasons to learn commonalities between these actions. On the other hand,

from a specific reason like "ladder broke", the model can learn very concise even if limited information, which applies to very specific actions.

Data Annotation Challenges. During the data annotation process, the workers had the choice to write comments about the task. From these comments we found that some difficulties with data annotation had to do with actions expressed through verbs that have multiple meanings and are sometimes used as figures of speech. For instance, the verb "jump" was often labeled by workers as "jumping means starting" or "jumping is a figure of speech here." Because the majority of videos containing the verb "jump" are labeled like this, we decided to remove this verb from our initial list of 25 actions. Another verb that is used (only a few times) with multiple meanings is "fall" and some of the comments received from the workers are: "she mentions the season fall, not the action of falling," "falling is falling into place," "falling off the wagon, figure of speech." These examples confirm how rich and complex the collected data is and how current state-of-the-art parsers are not sufficient to correctly process it.

4.7 Conclusion

In this chapter, we addressed the task of detecting human action reasons in online videos. We explored the genre of lifestyle vlogs, and constructed WHYACT – a new dataset of 1,077 video-clips, actions and their reasons. We described and evaluated several textual and visual baselines and introduced a multimodal model that leverages both visual and textual information.

We built WHYACT and action reason detection models to address two problems important for the advance of action recognition systems: adaptability to changing visual and textual context, and processing the richness of unscripted natural language.

The work in this chapter has been published in the 2021 Conference on Empirical Methods in Natural Language Processing. The dataset and the code introduced in this chapter are publicly available at https://github.com/MichiganNLP/vlog_action_reason.

In the next chapter, we propose to use the textual and visual information from lifestyle vlogs for another commonsense knowledge task: human action co-occurrence identification

▾ Instructions

You are given a video that contains a person describing an action and a list of **candidate reasons for why they want to do the action.**

From the list of candidate **reasons**, select the ones that are mentioned verbally or shown visually in the video.

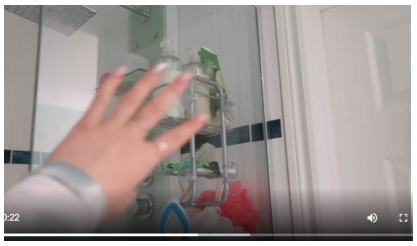
Please see three examples below:

1. **Action reasons are mentioned verbally, and shown visually in the video**

Answers:

- ☑ remove dirt (because it shown and metioned in the video)
- ☑ don't like dirtiness (because it is mentioned in the video)
- ☐ declutter
- ☐ company is coming
- ☐ feel productive

2. **Action reasons are mentioned verbally, but not shown visually in the video**



Answers:

- ☑ remove dirt (because it is mentioned in the video)
- ☐ don't like dirtiness
- ☐ declutter
- ☐ company is coming
- ☐ feel productive

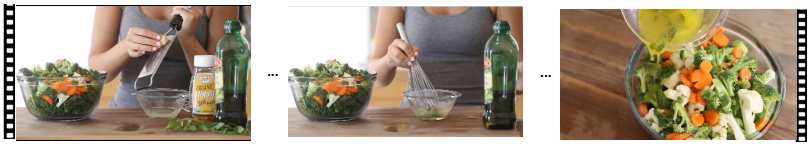
3. **Action reasons are shown visually, but not mentioned verbally in the video**

Answers:

- ☑ remove dirt (because it shown in the video)
- ☐ don't like dirtiness
- ☐ declutter
- ☐ company is coming
- ☐ feel productive

Figure 4.3: Instructions and examples of completed assignments with explanations for why the answers were selected.

Why is the person eating ?



"and I am gonna dress this I would say I do not I dress light to medium because other than that you do feel like you are eating plants by itself so I whisk the olive oil into the ingredients so it is nice to have a good rich dressing and this one was really delicious"

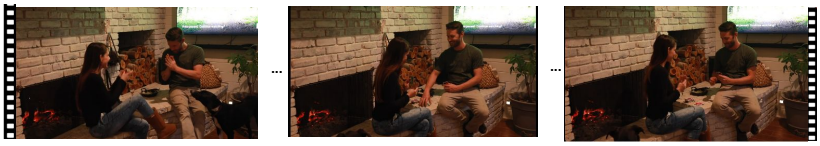
- ☑ tastes good
- × were hungry
- × had craving
- ☑ be healthy
- ☑ enjoy

Why is the person learning ?

"there is a bunch of farm courses that i wanted to take but i have been taking my time because i am busy with other things as well but i have been finding it interesting learning a lot and also trying to hone my skills .."

- ☑ know more information
- ☑ improve yourself
- ☑ fun
- ☑ become educated
- × course was recommended
- × avoid repeating mistakes

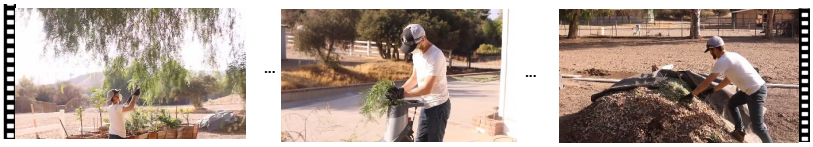
Why is the person playing ?



"it is fun because both of us are kind of competitive so it is always fun to play a board game or a card game so tonight we are playing uno normally but when our kids go to sleep we very often enjoy a shower or a bath together i share how much i enjoy my bathtub and my shower ..."

- ☑ win game
- × play music
- × bored
- × relax
- ☑ entertain yourself

Why is the person working ?



"... and as you can see he is actually working on extending our fence line and he is been doing that all on his own as well he also has quite a bit that he does on the tractor as well in order to keep the entire property clean ..."

- × have to
- ☑ complete job
- ☑ feel productive
- × need money

Why is the person painting ?

"... I wanted to show you how I made a little hanging burlap sign for my door that says Happy Easter now I have seen these done with Easter bunnies but I wanted to do a cross and this is what I am going to use to paint the cross on the burlap ..."

- × clean walls
- ☑ DYI craft project
- ☑ express yourself
- ☑ enhance appearance
- ☑ feel creative
- × change colors in home

Figure 4.4: Other examples of actions and their annotated action reasons in our dataset.

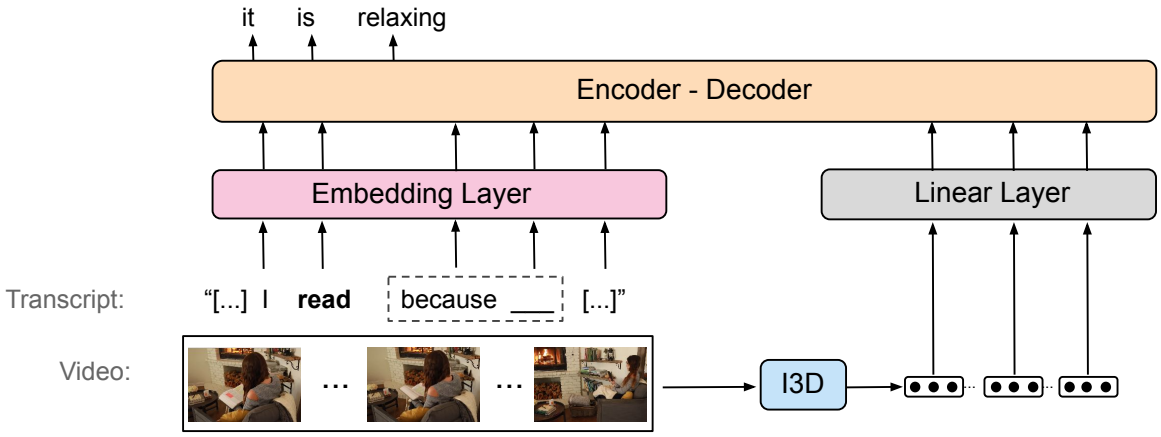


Figure 4.5: Overview architecture of our Multimodal Fill-in-the-blanks model. The span of text "because _____" is introduced in the video transcript, after the appearance of the action. This forces the T5 model to generate the words missing in the blanks. We then compute the probability of each potential reason and take as positive those that pass a threshold.

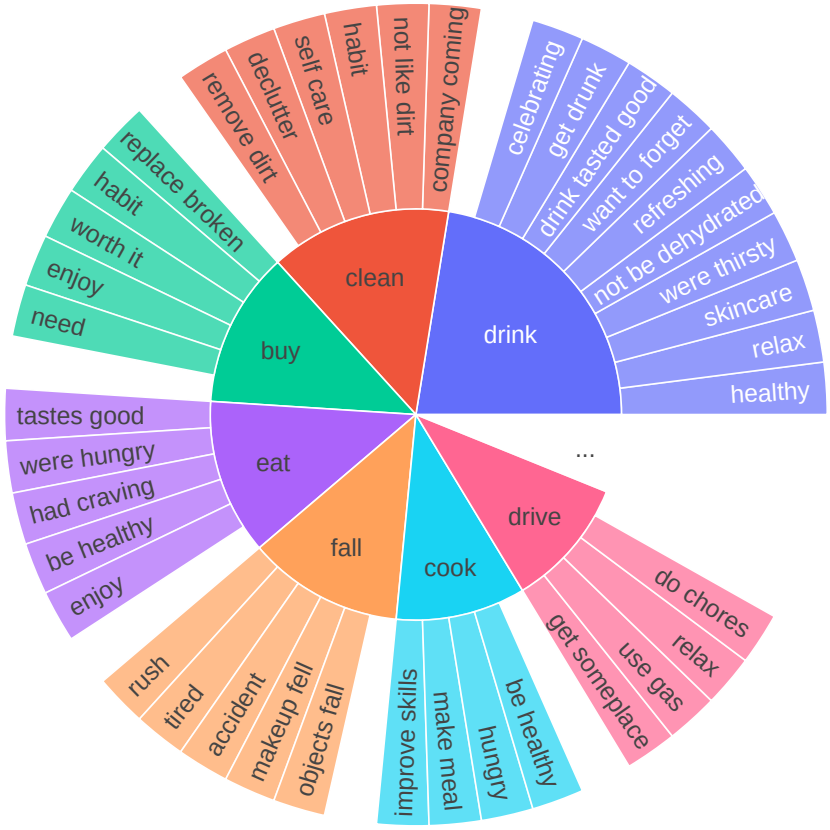


Figure 4.6: Distribution of the first seven actions, in alphabetical order, and their reasons, in our dataset.

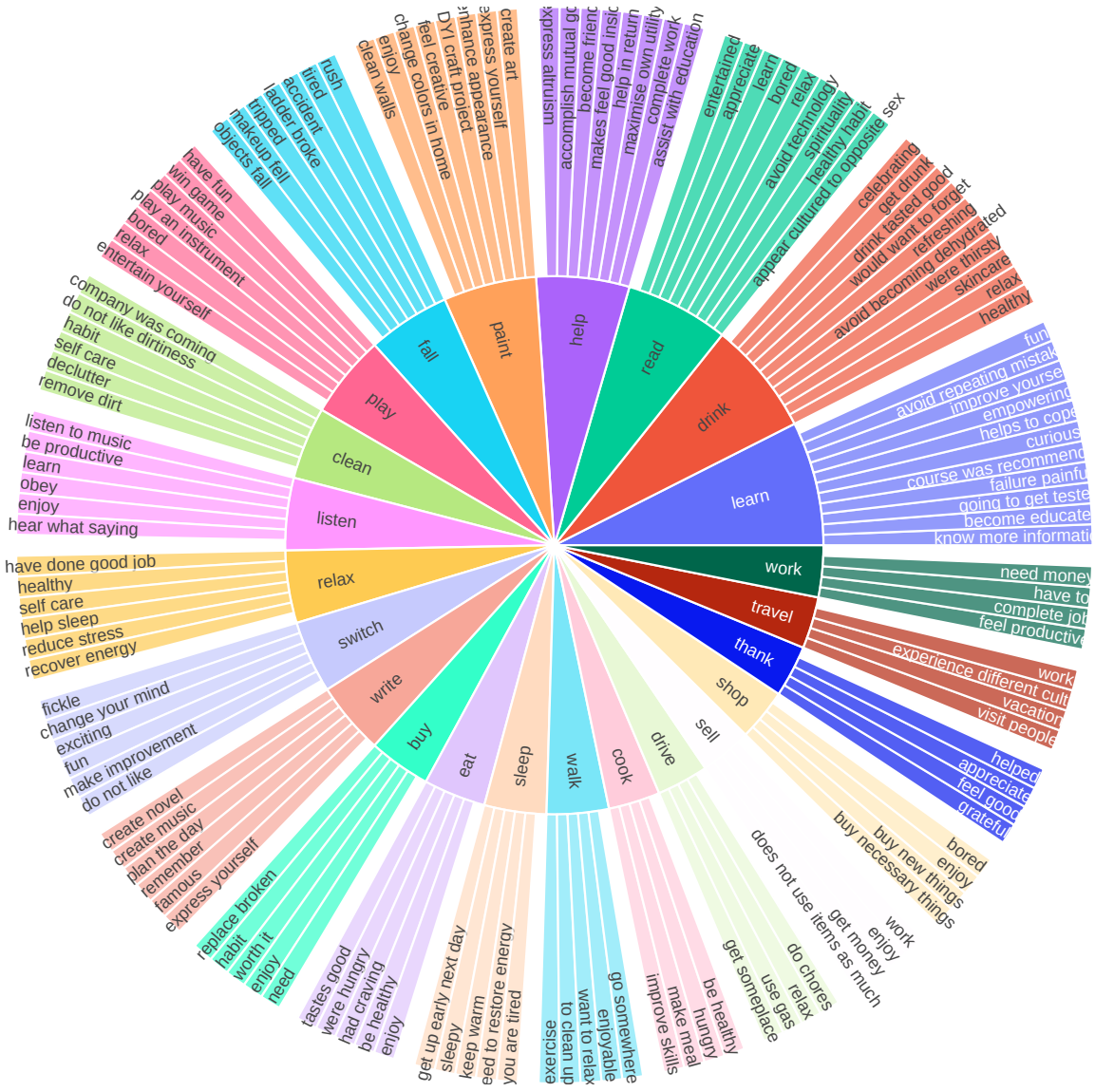


Figure 4.7: Distribution of all the actions and their reasons in our dataset.

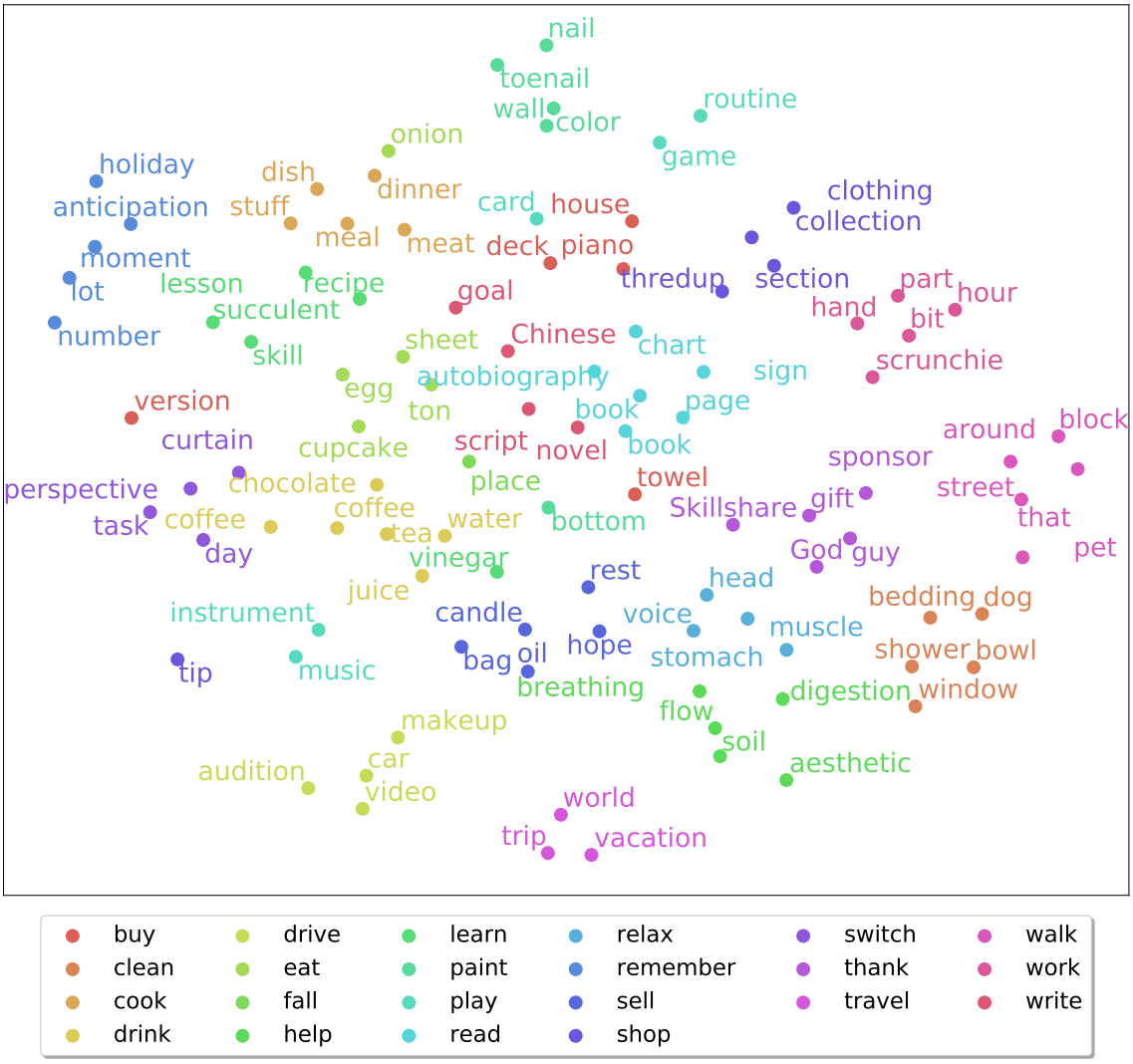


Figure 4.8: The t-SNE representation of the five most frequent direct objects for each action/verb in our dataset. Each color represents a different action.

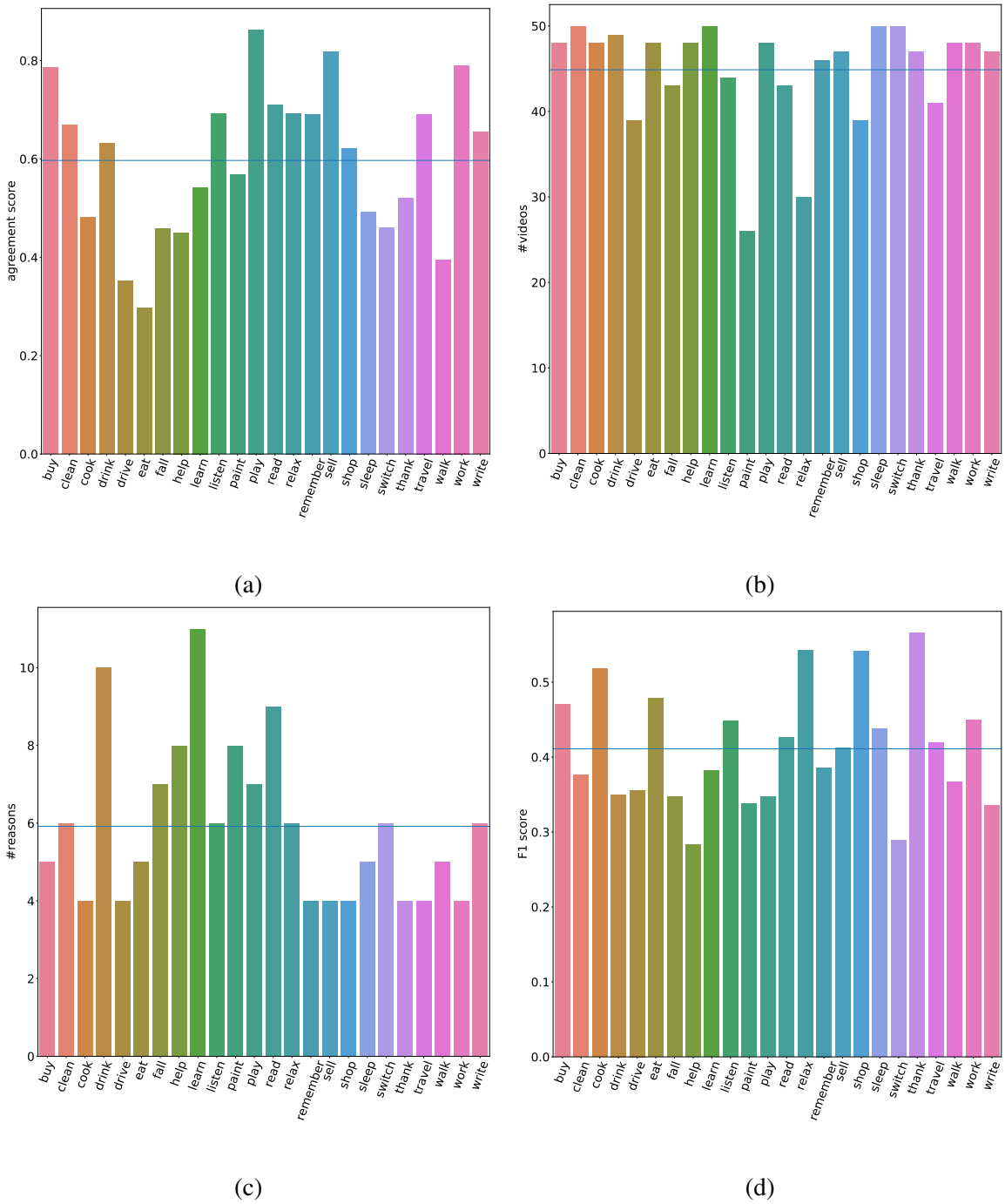


Figure 4.9: Distributions of all actions and their: (a) worker agreement score: Fleiss kappa score; (b) number of videos; (c) number of reasons; (d) F1 score obtained with the highest performing model (Fill-in-the-blanks with Text)

CHAPTER 5

Human Action Co-occurrence Identification

5.1 Introduction

Despite recent success in video human action recognition [11, 134, 135, 136, 137, 138], its applicability is still limited and error-prone once the application space is opened to everyday life. This indicates that current systems rely mostly on pattern memorization and do not effectively understand the action, which makes them fragile and unable to adapt to new settings [141, 142]. As a step toward enabling systems to gain more in-depth knowledge about human actions, we propose a new action understanding task: learning which actions are likely to occur in the same time interval, i.e., human action co-occurrence in everyday life scenarios.

Most human actions are interconnected, as an action that ends is usually followed by the start of a related action and not a random one (e.g., after "waking up", one would "wash face" or "make breakfast" and not "sell books" or "go to bed"). We model this information through co-occurrence relations: in general, we expect that the actions 'wake up", "wash face" and "make breakfast" co-occur in a short interval of time, while "wake up", "clean house" or "go to bed" do not. Current action recognition systems do not make use of this valuable information and treat each action independently. However, action recognition systems have the potential to become significantly more efficient and accurate if they had access to this kind of knowledge. For example, a system that just recognized the action "wake up" would have a high expectation that the following action is "make breakfast" or "wash face" and not "sell books" or "go to bed", and therefore it would discard such unrelated actions as not applicable.

The interconnection of human actions is very well depicted in lifestyle vlogs, where vlogger visually record their everyday routine consisting of the activities they perform during a regular day [4, 90, 175]. We collect a dataset of lifestyle vlogs from YouTube that reflect daily scenarios and are currently very challenging for systems to solve.

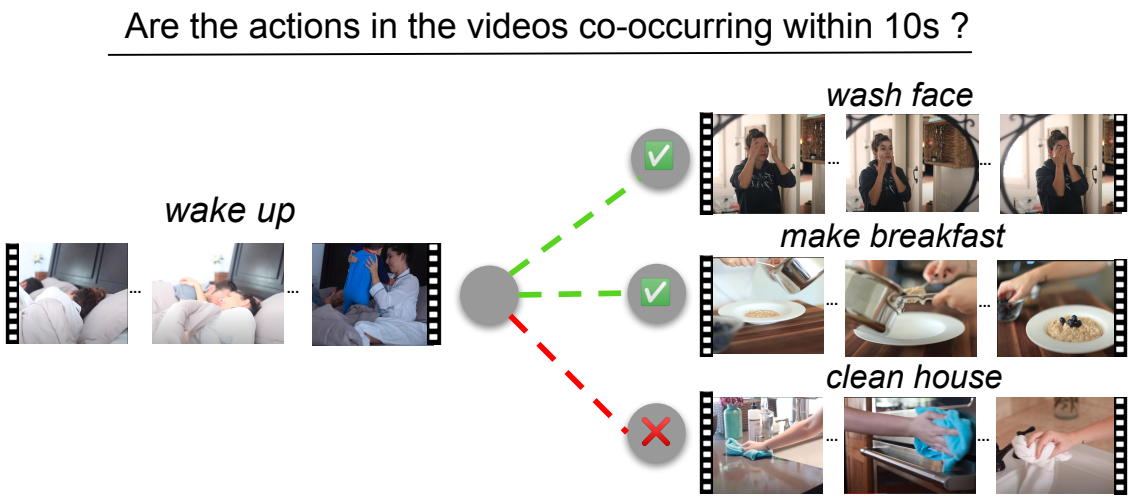


Figure 5.1: Human action co-occurrence in lifestyle vlogs: two actions co-occur if they occur in the same interval of time (10 seconds) in a video. The actions are represented as nodes in a graph, the co-occurrence relation between two actions is represented through a link between the actions, and the action co-occurrence identification task as a link prediction task.

A natural way to model the connections between human actions is through an undirected graph representation, where actions are represented as nodes, and their co-occurrences are represented as edges (Figure 5.1). An important advantage of this representation is that it reflects the transitive property of co-occurrence: if an action A co-occurs with action B, which in turn co-occurs with action C, action A, and action C are more likely to co-occur with one another than e.g., another action Z that does not co-occur with any of the actions A or B.

Another important advantage, which we leverage in this work, is that within this graph representation, the human action co-occurrence identification task can be formulated as a link prediction problem. We apply simple but powerful topology heuristics and learning models that use the graph representation to capture novel and useful information about human actions, and we show that this formulation leads to significant improvements in action co-occurrence identification as compared to models that handle one action at a time.

This work makes four main contributions. First, we formalize the new task of human action co-occurrence identification in online videos. Second, we introduce a new dataset, ACE, consisting of a large graph of co-occurring actions in online vlogs. Third, we propose several models to solve the task of human action co-occurrence, by using textual, visual and multi-modal action representations. Finally, we also present an analysis of our new dataset, which leads to rich avenues for future work for improving the tasks of action co-occurrence and ultimately action recognition in online videos.

5.2 Related Work

There are three areas of research related to our work: human action co-occurrence, graph link prediction, webly-supervised learning

Human Action Co-occurrence. Recent work shows that action co-occurrence priors [176, 177] increase the performance of human-object interaction models and lead to more effective training, especially in rare classes. To our knowledge, this is the only work that explores the impact of action co-occurrence information on action recognition models. Unlike our work, they assume that the action co-occurrence information is provided and they do not attempt to learn it. To the best of our knowledge, we are the first to propose the task of learning human action co-occurrence in videos.

Human action co-occurrence identification is also related to learning action temporal order in videos which is used to construct the co-occurring action pairs. Misra et al. [178] propose the task of temporal order verification, i.e., to determine whether a sequence of frames from a video is in the correct temporal order. Using this simple task and no semantic labels, they learn visual representation. In our work, we learn action representations using the information extracted from the action co-occurrence graph, which is a more general relation reflecting a shared context among the actions.

Link Prediction. Link prediction is a key problem for graph-structured data and is relevant for our graph formulation of action co-occurrence. The objective of link prediction is to predict whether two nodes in a graph are likely to be linked [179].

Link prediction approaches can be categorized into three main categories [180]: similarity-based/heuristic [181, 182, 183, 184, 185, 186, 179]; probabilistic-based [187]; and dimensionality reduction-based (e.g., embedding-based or other learning approaches; [188, 189]).

For our task, we apply the similarity-based, embedding-based, and learning-based models. Similarity-based methods are the simplest and measure similarity between every pair of nodes using topology properties of the graph (e.g., common neighbors). The embedding-based link prediction models map the embedding of nodes to a lower dimension such that similar nodes have similar embeddings. The learning-based link prediction models can be cast using supervised classification models where a point corresponds to a node pair in the graph, and the point label represents the presence or absence of an edge/link between the pair. This is typically a binary classification task where several classifiers (e.g., decision tree, SVM, etc.) can be employed to predict the label of unknown data points.

Webly-Supervised Learning. In our work, we identify human action co-occurrence in the context of rich, virtually unlimited, constantly evolving online videos from YouTube, using the video transcripts as a web supervision signal. Large-scale video datasets [22] on instructional videos and lifestyle vlogs [4, 90, 175] are other examples of web supervision. The latter is similar to our work as they analyze online vlogs, but unlike our work, their focus is on action detection or the reasons behind actions and not on action co-occurrence.

5.3 Dataset

In order to develop and test models for determining if two actions are co-occurring, we compile a novel dataset, which we refer to as ACE (Action Co-occurrencE).

5.3.1 Data Collection

We start by compiling a set of lifestyle videos from YouTube, consisting of people performing their daily routine activities, such as cleaning, cooking, studying, relaxing, and others. We build a data gathering pipeline to automatically extract and filter videos and their transcripts.

We select 20 YouTube channels and download all the videos and their transcripts. The channels are selected to have good quality videos with automatically generated transcripts containing detailed verbal descriptions of the actions depicted.

An analysis of the videos indicates that both the textual and visual information are rich sources for describing not only the actions but also in what order the actions are performed, making them a great source of data for developing action co-occurrence models. The routine nature of the videos means that the vloggers record and describe their actions in the order they normally occur in a day: e.g., "wake up", "make bed", "wash face", "make breakfast", "drive to work", and so on. They can also choose to focus on certain activities (e.g., often cooking) and enumerate more fine-grained actions related to those activities (e.g., "cut apple", "add peanut butter"). Therefore, our dataset contains both general and fine-grained actions. We present qualitative and quantitative analyses of our data in Section 5.6.

Action extraction. Having a comprehensive list of actions is necessary for creating graphs that contain most of the actions in the videos. At the same time, not all the actions from the transcripts are useful, as many of them are not visible in the video or hard to detect by computer vision systems (e.g., "feel","talk", "thank", "hope", "need", "see', "try").

Therefore, we first make sure that the actions we collect are mostly visible in the videos. Our strategy is to extract all the verbs from the transcripts and then filter them using a list of "visual verbs" collected from imSitu [190], COCO-a [191] and Levin [192].[1] Verbs from imSitu and COCO-a are considered visual as the dataset collection pipelines include an explicit annotation step to determine if verbs are visual. We manually filter and check the verbs collected from Levin.

Next, we extract all actions from the video transcripts using the dependency parser from spaCy [162] by extracting all the verbs and their corresponding verb phrase direct objects, prepositions, and objects of prepositions. We find that extracting only verbs and their corresponding direct objects does not always return comprehensive actions (e.g., "add teaspoon" versus "add teaspoon of salt"). We also find that many verbs do not have informative direct objects (e.g., "write it", "clean them"), which makes the actions harder to differentiate and visually recognize. To address this, we apply co-reference resolution on the video transcripts using spaCy [162] NeuralCoref[2] model, and re-extract the actions from the processed transcripts.

Finally, we obtain our visible actions by filtering all the transcript extracted actions that contain visual verbs.

Video extraction. As transcripts are temporally aligned with videos, we can obtain meaningful video clips related to the narration. We extract clips corresponding to the visual actions based on transcript timestamps. From 2,571 videos, we obtain 19,685 unique video clips and 25,057 (action, video-clip) pairs. Note that an action can be present in multiple video clips, and conversely, a video clip can contain multiple actions. To control the number of clips per action, we randomly sample up to 10 random video clips for each action and finally obtain 12,994 (action, video-clip) sampled pairs.

5.3.2 Data Pre-processing

After collecting the videos, transcripts, and visual actions, the following data pre-processing steps are applied.

Action Co-occurrence. From all the extracted visual actions, we automatically select all the action pairs that are co-occurring. We define two actions as co-occurring if they are

[1] Levin's taxonomy provides a classification of 3,024 verbs (4,186 senses) into 48 broad and 192 fine-grained classes. We leave analyzing the Levin verb taxonomy impact on human action model performance as a future work direction.

[2] https://spacy.io/universe/project/neuralcoref

	#Verbs	#Actions	#Action pairs
Initial	608	20,718	-
Co-occurrence	439	18,939	80,776
Clustering	172	2,513	48,934
Graph	164	2,262	11,711

Table 5.1: Statistics for the collected number of unique verbs, actions, and co-occurring action pairs at each stage of data pre-processing.

less than 10 sec away from each other. The 10 sec is an intermediate value threshold we set after experimenting with other values (5 and 15 sec). The lower the threshold, the fewer co-occurring action pairs we will extract and vice-versa. Note that the threshold controls the scale of time we choose to focus on when collecting co-occurring actions: e.g., only short actions (e.g., "open fridge", "get milk") might be captured in a small interval of time, while longer intervals allow for longer actions to co-occur (e.g., "cook meal"). We choose an intermediate value that allows for both shorter and longer actions to co-occur. Note that the captured actions depend also on the filming style (e.g., vloggers could increase the filming time of normally short actions).

For computing the distance in time between two actions, we use the transcript time stamps. Note that we use the time the actions are mentioned in the transcript, which is not always aligned with the time the action visually appears in the video. However, we find that the transcript time stamps are sufficient for our task, as the actions mentioned in the transcript usually follow the order from the video, and some misalignment is mediated by filtering out motionless videos and by collecting multiple videos per action.

Action Clustering. We find that many actions are often very similar in meaning. This leads to many action repetitions: e.g., "use iron", "iron shirt", "iron cloth". In order to avoid action repetitions, we group similar actions by clustering all actions. We represent each action using the pre-trained model Sentence-BERT [166] and apply Agglomerative Clustering [161]. We filter out the clusters of actions that contain less than 2 actions, as they are likely to be outliers/actions that were not well extracted. The actions in each cluster are then renamed to the most common action in the cluster: e.g., "iron shirt", "iron cloth" are renamed to "use iron".

We observe that the clustering model is introducing some level of noise as it does not perfectly cluster all actions. We tried to mitigate this by experimenting with different Sentence-BERT pre-trained models for sentence similarity[3] and fine-tuning our clustering

[3] sbert.net/docs/pretrained_models.html

model hyper-parameters[4] based on automatic evaluation metrics for measuring the quality of clusters: Silhouette Coefficient [193], Calinski-Harabasz Score [194], and Davies-Bouldin Index [195].

Action Graph Filtering. After we rename the actions based on clustering, we create a graph, where the graph nodes represent the actions and the graph edges represent the relations between two actions. Specifically, we create an undirected graph for each video, where the graph nodes are represented by the actions in the video, and the co-occurring actions are connected by an edge. Each edge has weight, which is equal to the number of times the corresponding actions co-occur in the video.

We combine all the video graphs to obtain a single large graph that contains all the co-occurring actions in our data. We filter out the action pairs that co-occur only once in the graph (their edge weight is equal to one), as their co-occurrence relation is not strong and might be random.

We show the statistics before and after all the action filtering steps in Table 5.1. More information about our dataset (e.g., action frequency distributions, most common actions, action pairs) can be found in Section 5.6.

5.4 Identifying Action Co-occurrence in Vlogs

We formulate our action co-occurrence identification task as a link prediction task. Link prediction aims to predict the existence of a link between two nodes in a graph. In our setup, nodes are represented by actions, and every two co-occurring actions are connected by a weighted edge, where the weight represents the number of times the two actions co-occur. Our goal is to determine if there exists an edge between two given actions.[5]

5.4.1 Data Representation

Textual Representations. To represent the textual data – actions and their transcript context, we use Sentence Embeddings computed using the pre-trained model Sentence-BERT embeddings [166] calculated using the graph topology and the textual embeddings obtained from CLIP [196]. When computing CLIP textual action embeddings, we concatenate the action with given prompts (e.g., "This is a photo of a person"), as described in the original paper [196].

[4] linkage distance threshold (1.5), linkage criterion (ward)

[5] At this point, we do not aim to also identify the strength of the link.

Video Representations. We use the CLIP model [196] to represent all the actions and their corresponding video clips. One action can have multiple video clips: an action has at most 10 corresponding videos. From each video clip, we extract four equally spaced frames and pre-process them as described in their paper [196]. We use the pre-trained Vision Transformer model ViT-B/16 [197] to encode the video frames and the textual information. We apply the model to each of the four frames and average their representations [198].

Graph Representations. We also use the training graph topology information (node neighbors and edge weights) to compute action embeddings as the weighted average of all of their neighbor node embeddings, where the weights are edge weights (i.e., how many times the two nodes co-occur). The neighbour node embeddings are represented using either textual embeddings (Sentence-BERT; [166]) or visual embeddings (CLIP; [196]). All the graph-based models described in the next section use graph topology information from the validation graph (see Section 5.5.1).

We use the representations described above as input to different action co-occurrence models.

5.4.2 Action Co-occurrence Models

We explore different models with different input representations. We group the models as described in the related work link prediction section: random baseline, heuristic-based models (graph topology models), embedding-based models (cosine similarity and graph neural networks), and learning-based models (SVM models). As described in Section 5.4.1, we run experiments with various types of data representations: Textual: Action and Action Transcript; Visual: Action, Video, and Multi-modal (Action&Videos) (the average between action and video visual embeddings); Graph: Action and Multi-modal (Action&Videos) using the graph topology.

5.4.2.1 Random Baseline

The action pairs to be predicted as co-occurring or not are split into equal amounts, therefore a random baseline would have an accuracy score of 50%.

5.4.2.2 Heuristic-based Graph Topology Models

We apply nine popular node similarity methods that only use graph topology information in the prediction process: Common Neighbours [181], Jaccard Index [182], Salton Index [183]), Preferential Attachment [199], Adamic-Adar Index [184], Hub Depressed Index,

Hub Promoted Index [185], Resource Allocation [186], and Shortest Path [179]. Note that the heuristic-based methods do not use any of the data representations described in 5.4.1. We describe each of the methods above:

Notation. Let s_{xy} be the similarity between nodes x and y, $\Gamma(x)$ be the number of nodes connected to node x and k_x be the degree of node x.

Common Neighbours. Two nodes are more likely to be connected if they have more common neighbors.

$$s_{xy} = |\Gamma(x) \cap \Gamma(y)| \tag{5.1}$$

Jaccard Index. Measures the proportion of common neighbors in the total number of neighbors.

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{5.2}$$

Salton Index. Measures the cosine of the angle between columns of the adjacency matrix, corresponding to given nodes.

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{k_x k_y}} \tag{5.3}$$

Preferential Attachment. Preferential attachment means that the more connected a node is, the more likely it is to receive new edges.

$$s_{xy} = k_x k_y \tag{5.4}$$

Hub Promoted Index. This measure assigns higher scores to edges adjacent to hubs (high-degree nodes), as the denominator depends on the minimum of the degrees of the nodes of interest.

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\min\{k_x, k_y\}} \tag{5.5}$$

Hub Depressed Index. This measure, in contrast to Hub Promoted Index, assigns lower scores to edges adjacent to hubs. It penalizes large neighborhoods.

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\max\{k_x, k_y\}} \tag{5.6}$$

Adamic-Adar Index. This measure counts common neighbors by assigning weights to nodes inversely proportional to their degrees. That means that a common neighbor, which is unique for a few nodes only, is more important than a hub.

$$s_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log k_z} \tag{5.7}$$

Resource Allocation. Measures how much resource is transmitted between two nodes.

$$s_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z} \tag{5.8}$$

Shortest Path. The similarity score is inversely proportional to the length of the shortest path between two nodes.

$$s_{xy} = \frac{1}{min\{l : path_{xy}^{<l>} exists\}} \tag{5.9}$$

Weighted Graph Models. Our graph is weighted, therefore we also apply weighted graph models. We modify some of the above models (Common Neighbours, Adamic-Adar Index, and Resource Allocation), to use the link weight information, as described in the work from Zhu and Xia [200]. We find that, for our graph, using the weight of the links achieves similar results as without them.

5.4.2.3 Embedding-based Models

Cosine Similarity. To determine if two given actions co-occur, we compute the cosine similarity between their corresponding embeddings. If the similarity score is greater than a threshold, fine-tuned on validation data, we predict the actions as co-occurring.

Graph Neural Networks. We also use Graph Neural Network (GNN) models. We choose four diverse and popular GNN models [180]: node2vec [188], attri2vec [201], GraphSAGE [202], and GCN [189].

Graph Neural Network models can also be classified as learning-based models: they learn a new heuristic from a given network, as opposed to Graph Topology models, which use predefined heuristics, i.e., score functions. We create our graph based on a known heuristic: co-occurring actions are closely connected in the graph. Therefore, we hypothesize that heuristic models will perform better. Indeed, we observe that for our graph, the GNN methods do not perform better than the heuristic models: the best performing model is GCN with 79.8% accuracy, while the best performing topology model has an 83.3% accuracy (see Table 5.2). Therefore, we conclude that our task does not benefit from these advanced neural models.

5.4.2.4 Learning-based Model

We run a support vector machine (SVM) [43] classifier on each of the action pairs to be classified as co-occurring or not. We concatenate all the input representations/ embeddings and all the heuristic scores, we standardize the features by removing the mean and scaling to unit variance. We fine-tune the model hyper-parameters (kernel type, C, gamma) on the validation data, using a grid search.

5.5 Evaluation

We conduct extensive experiments to evaluate the action pairs co-occurrence identification task. The task can be represented as a graph link prediction task. Therefore, we adopt the link prediction evaluation process.

5.5.1 Evaluation Data Split

We split the original graph into train, validation, and test graphs. We use the train graph for training the models, the validation graph for fine-tuning the hyper-parameters, and the test graph for evaluating the model's performance on held-out data.

In link prediction, the goal is to predict which links will appear in the future of an evolving graph. Therefore, while keeping the same number of nodes as the original graph, the number of edges is changed as some of the edges are removed during each split and used as the positive samples for training, fine-tuning, and testing the link prediction models. The edges are split into train, validation, and test sets using a transductive split, which is considered the default evaluation splitting technique for link prediction models.[6] More specifically, we randomly sample 10% of all existing edges from the original graph as

[6] http://web.stanford.edu/class/cs224w/

positive testing data and the same number of nonexistent edges (unconnected node pairs) as negative testing data. The reduced graph becomes the test graph and together with the set of sampled edges is used for testing the models. We repeat the same procedure to obtain the validation and the training data for the models. The validation graph is obtained by reducing the test graph, and the training graph is obtained by reducing the validation graph.

Model	Accuracy
BASELINE	
Random	50.0
HEURISTIC-BASED	
Common Neighbours	81.2
Jaccard Index	80.2
Salton Index	82.5
Preferential Attachment	74.7
Hub Promoted Index	83.3
Hub Depressed Index	81.1
Adamic-Adar Index	82.7
Resource Allocation	78.8
Shortest Path	82.9
EMBEDDING-BASED	
Cosine similarity	82.8
node2vec	77.2
attri2vec	78.4
GraphSAGE	78.3
GCN	79.8
LEARNING-BASED	
SVM	**91.1**

Table 5.2: Accuracy results for all the models.

5.5.2 Results and Ablations

Table 5.2 contains the results, measured by accuracy, for each model type. The learning-based model, SVM, using all input representations (textual, visual, graph) and all graph heuristic scores obtains the highest accuracy score. Therefore, using both graph topology information and textual embeddings leads to the best performance for our task.

The results for each of the heuristic-based, graph topology models are shown in Ta-

Model	INPUT REPRESENTATIONS						
	Textual		Visual			Graph	
	Action	Transcript	Action	Video	Action&Video	Action	Action&Video
Similarity	60.6	65.2	62.7	57.0	65.4	**82.8**	50.6
SVM	76.3	71.1	73.1	76.2	76.1	80.9	74.6

Table 5.3: Ablations and accuracy results on test data. We compute the ablations for each input representation: textual, visual, and graph, for an embedding-based model (cosine similarity) and a learning-based model (SVM); the heuristic-based models do not depend on input representation type, therefore we do not ablate them.

ble 5.2. Simple heuristics (common neighbors or length of the shortest path) are enough to achieve a good performance on our graph.

The ablation results, split by input representation are shown in Table 5.3. We analyse how different input representations influence the models performance: textual (Sentence-BERT and CLIP textual) vs. visual (CLIP visual) vs. multi-modal (CLIP textual and visual) vs. graph (Sentence-BERT and CLIP textual and visual). The input representations are described in Section 5.4.1. The textual embeddings are a strong signal for our task, even when not using any graph information: SVM with only Action Sentence-BERT embeddings has a 76.3% accuracy. Using graph representations or graph heuristic information leads to better performance (80.9% and 91.1% accuracy, respectively). The visual and multi-modal embeddings are also valuable but perform worse compared to the textual embeddings. We hypothesize that CLIP embeddings might not work very well for our task for two main reasons. First, the video clips contain some amount of misalignment with the actions, which affects CLIP performance (as typical for vlogs, the time the action is mentioned is not the same as the time it is shown in the video, and other actions might be shown in the same video-clip). However, the visual modality offers important information about human actions and can be used in future work with better, more robust visual models.

5.5.3 Action Nearest Neighbours Retrieval

To show the usefulness of the action embeddings, we propose a downstream task: action nearest neighbors retrieval. We compare three action representations: textual (Action Sentence-BERT embeddings), visual (Video CLIP embeddings), and graph-based (graph weighted average of neighbor nodes Action Sentence-BERT embeddings). In Figure 5.2 we show the top three action neighbors, from each of the three representations, for three random actions from our dataset. We observe that each representation captures different

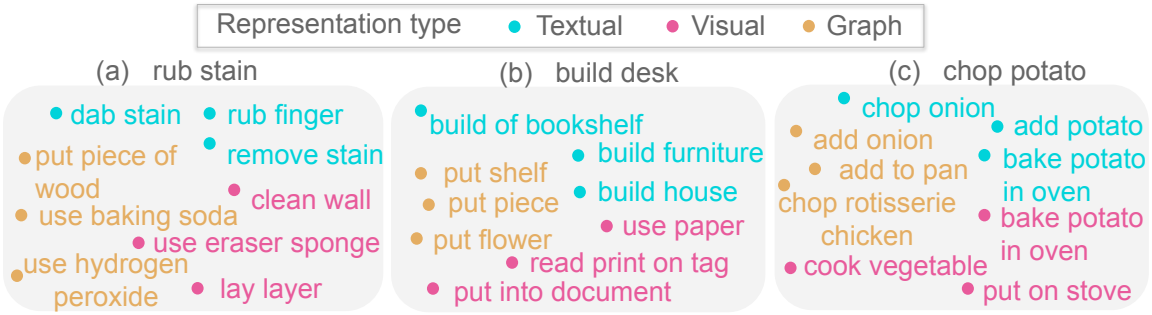


Figure 5.2: Top three action neighbors, from each of the three representations, for three random actions from our dataset: "rub stain", "build desk", "chop potato". The neighboring actions are shown in different colors, based on their embedding type. Best viewed in color.

kinds of information: the actions obtained with textual representations are more syntactically similar to the original action, sharing either the verb or the object (e.g., "chop potato": "add potato", "chop onion"), while the actions obtained with visual and graph representations are more diverse and capture "location" information (actions expected to be temporally closely depicted in a video) (e.g., "chop potato": "put on stove", "add to pan").

5.6 Data Analysis

We want to determine which actions co-occur the most in our dataset. This knowledge is valuable for action recognition and action prediction systems. Systems enriched with this knowledge can make more informed decisions when predicting or recognizing actions. Specifically, action recognition systems can discard actions that are unlikely to happen given a previous action and assign a higher probability to the actions that are known to co-occur with the previous action (e.g., given that a system previously recognized the action "wake up", a next likely action could be "wash face", and not "clean house").

Given two actions, we compute their co-occurrence score using the Positive Pointwise Mutual Information (PPMI) [203]. PMI is biased towards infrequent words, therefore we do not compute PMI for infrequent actions (that appear less than 10 times).

$$PPMI_{a_i,a_j} = \max(\log \frac{P_{a_i,a_j}}{P_{a_i} P_{a_j}}, 0) \tag{5.10}$$

$$P_{a_i,a_j} = \frac{\#(a_i, a_j)}{\#action\ pairs}, P_{a_k} = \frac{\#a_k}{\#actions} \tag{5.11}$$

Figure 5.3 shows the co-occurrence matrix between the top 20 most frequent actions. The most frequent actions are related to cooking and we can observe how actions related to adding ingredients are co-occurring between themselves (e.g. "add potato" and "add

avocado") or with actions related to adding something to a container (e.g."add potato" and "add to bowl"). Section 3.3.3 includes additional information: co-occurrence matrices of top 50 most frequent actions and verbs (Figures 5.5 and 5.6), top 15 actions and verb pairs that co-occur the most and the least (Section 5.6.1), actions distributions (Figure 5.7) and top 10 most frequent clusters (Figure 5.4).

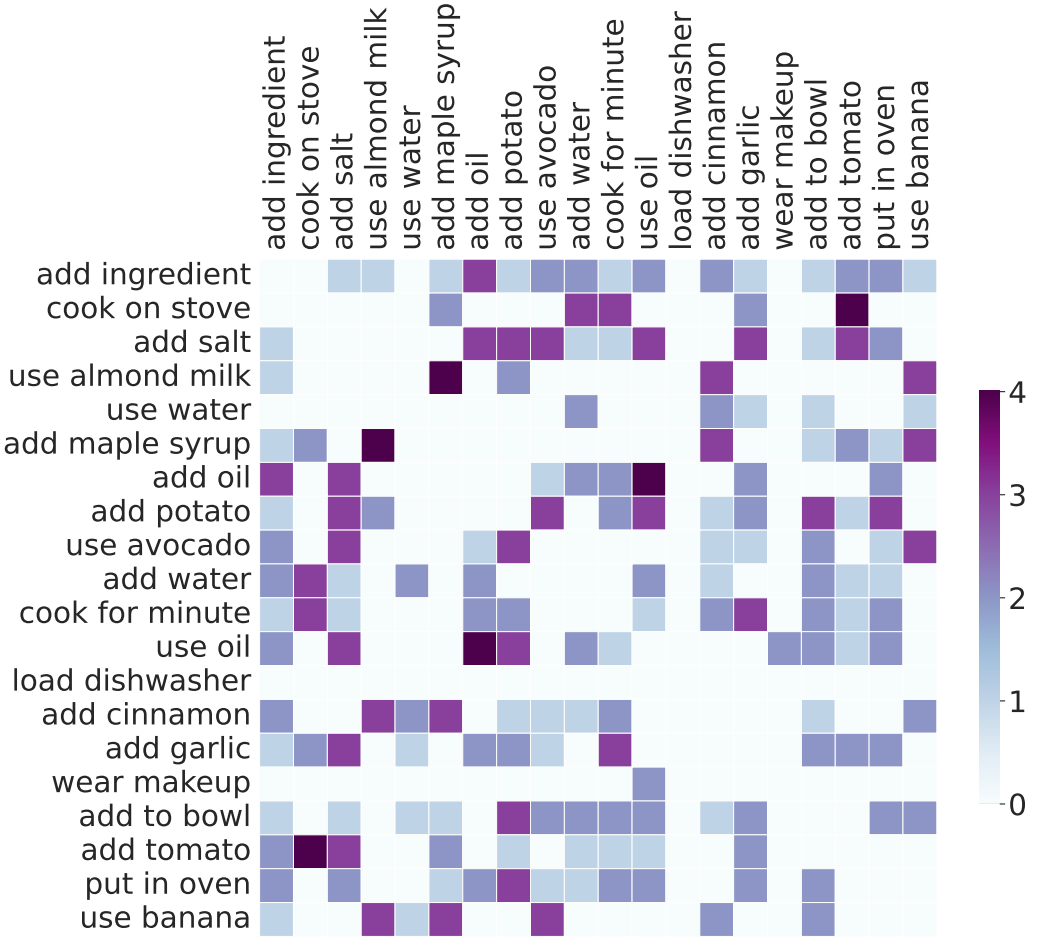


Figure 5.3: Co-occurrence matrix for the top 20 most frequent actions in our dataset, ACE. The scores are computed using the PPMI measure: actions with higher scores have a stronger co-occurrence relation and vice-versa. For better visualization, we sort the rows of the matrix to highlight clusters. Best viewed in color.

5.6.1 Action Clustering

Recall that all the raw actions extracted from the transcript are clustered as described in Section 5.3.2. In order to analyze the content of the clusters, we show the 10 most frequent clusters using t-distributed Stochastic Neighbor Embedding (t-SNE) [174] (see Figure 5.4)

Figure 5.4: The t-SNE representation of the ten most frequent action clusters in our dataset. Each color represents a different action cluster. Best viewed in color.

By examining the clusters, we can distinguish some open challenges or future work directions. First, there are multiple ways of expressing the same action, which can be seen when looking at the actions inside each cluster (e.g., “add to bowl’, “add into bowl”, “place in bowl”, “use measuring bowl”). This showcases the complexity of language. Second, the cluster algorithms are not perfect and some clusters could be merged (e.g., “add water” and “use water”) or some actions should not belong in some of the clusters (e.g., “put engine oil” and “paint with oil”). Third, actions can be too ambiguous (“use water”) or too broad (e.g., “add ingredient”).

Action pair	Frequency	Verb pair	Frequency
load dishwasher, wash dish	52	add, use	3864
eat food, eat in day	29	use, use	2987
use shampoo, wash hair	26	add, add	2895
use cloth, use water	24	put, use	1786
add sweetener, add teaspoon of maple syrup	23	add, put	1060
use almond milk, use milk	22	add, cook	814
use butter, use purpose flour	22	clean, use	724
add olive oil, massage kale	22	put, put	620
load dishwasher, load dishwasher at night	22	use, wear	366
clean steel appliance, use cloth	21	add, chop	355
put dish, wash dish	19	clean, clean	330
clean toilet, spray toilet	19	cut, use	328
clean sink, use dish soap	19	use, wash	317
add cocoa powder, use purpose flour	17	add, eat	293
squeeze lemon juice, use lemon	17	cook, use	284
...	...	...	...
pack makeup bag with, put in ziploc bag	2	bake, pull	2
put on skin, use for lip	2	bake, stick	2
put stuff, use on cuticle	2	pack, pull	2
put under eye, use on cuticle	2	empty, hold	2
put on eyelid, use on cuticle	2	brush, mix	2
fill brow, use on cuticle	2	attach, paint	2
read book, use business card	2	pour, wrap	2
spray paint, use iron	2	fight, wash	2
use product, use vegetable peeler	2	drink, massage	2
teach responsibility, work in beauty industry	2	add, poke	2
use charcoal scrub, use scrub	2	stick, stir	2
use charcoal scrub, use vegetable peeler	2	fill, scrape	2
use charcoal scrub, use steamer	2	carve, cover	2
add tea to water, use charcoal scrub	2	curl, open	2
open pore, use charcoal scrub	2	curl, rinse	2

Table 5.4: Top 15 most and least frequent action pairs (left) and verb pairs (right) in our dataset.

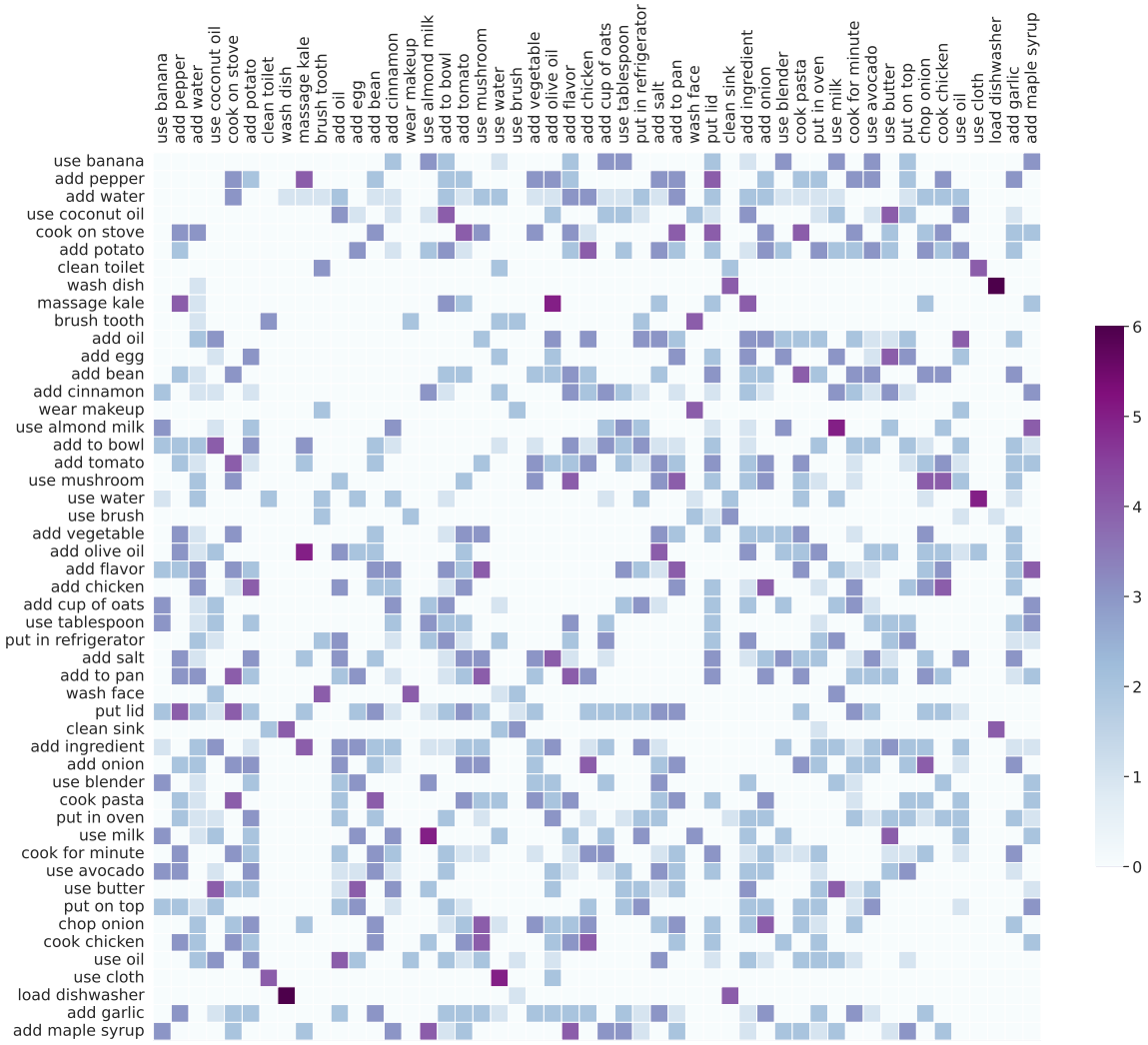


Figure 5.5: Co-occurrence matrix for the top 50 most frequent actions in our dataset, ACE. The scores are computed using the PPMI measure: actions with higher scores have a stronger co-occurrence relation and vice-versa. For better visualization, we sort the rows of the matrix to highlight clusters. Best viewed in color.

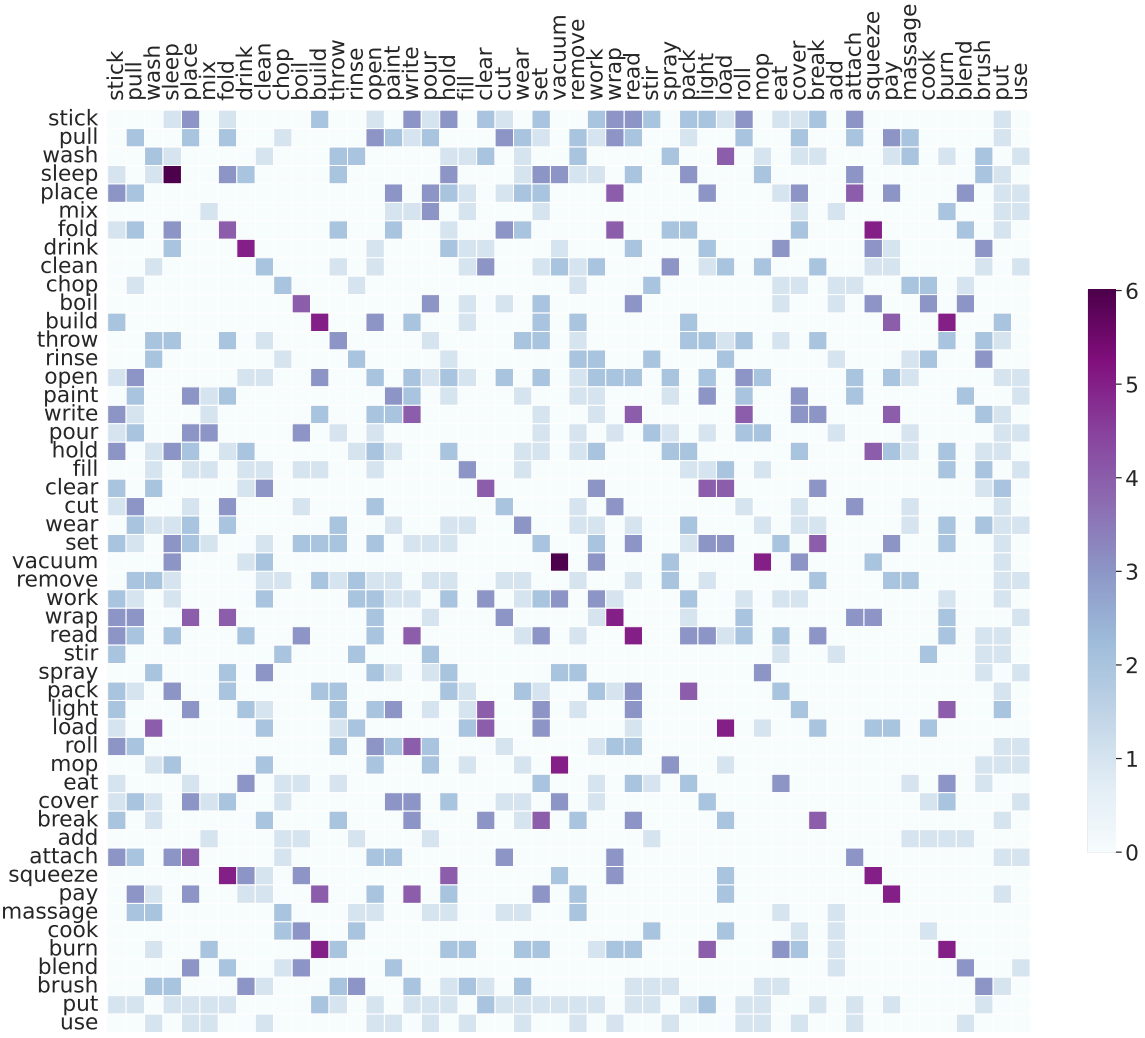


Figure 5.6: Co-occurrence matrix for the top 50 most frequent verbs in our dataset, ACE. The scores are computed using the PPMI measure: actions with higher scores have a stronger co-occurrence relation and vice-versa. For better visualization, we sort the rows of the matrix to highlight clusters. Best viewed in color.

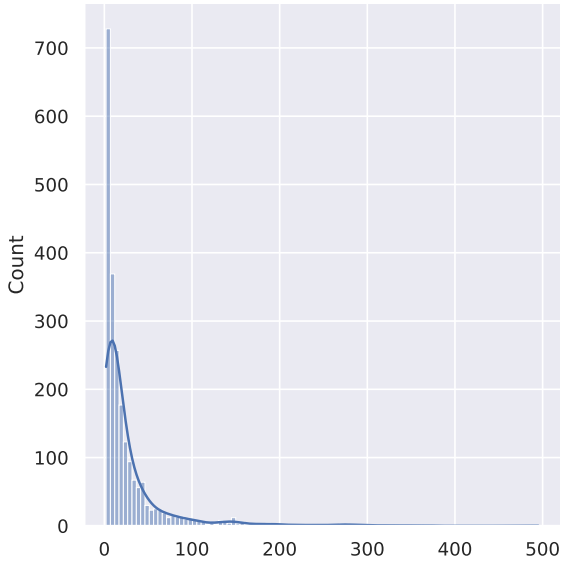


Figure 5.7: Action distribution in our dataset, ACE: count of actions frequencies.

5.7 Conclusion

In this chapter, we addressed the task of detecting co-occurring human actions in online videos. We explored the genre of lifestyle vlogs, and constructed ACE, a new dataset of ∼12k pairs of visual actions and their corresponding video clips. We described and evaluated models that leverage textual, visual, and graph information.

We built ACE and action co-occurrence identification models to address a task that can lead to advances in action recognition systems: capturing the relations between human actions. We are the first to address this problem and to use graph representations in this setting. We show that graph representations are useful for our task and capture novel information about human actions, which complements the representations learned from the current language and visual models.

In future work, we plan to experiment with our graph action representation in action recognition systems to improve their performance. The ACE dataset and the code introduced in this chapter are publicly available at https://github.com/MichiganNLP/vlog_action_co-ocurrence.

CHAPTER 6

Conclusions

Throughout this dissertation, we explored various aspects of human action understanding and leveraged multiple modalities such as the visual and textual channels. The goal of this thesis was to use such representations to develop computational models for human action understanding.

In particular, our approach used different representations to address four challenging problems, which we grouped into two categories: *physical*, i.e., that rely mostly on visual information and *commonsense*, i.e., that require both visual and context information. These two categories are complementary to each other and together aim to encompass human action understanding. The *physical* tasks we addressed are: i) human action visibility identification in online videos, ii) temporal human action localization in online videos, and the *commonsense* tasks we addressed are: iii) human action reason identification in online videos and iv) human action co-occurrence identification in online videos.

6.1 Research Questions Revisited

At the beginning of this thesis, several research questions were formulated, which were addressed in turn by the experiments described in the thesis. The findings of the thesis are summarized below.

1. **Are vlogs well suited for learning about human actions and behaviors?**

   In Chapter 2 we introduced a dataset of routine and do-it-yourself (DIY) videos from YouTube, consisting of people performing daily activities, such as making breakfast or cleaning the house. These videos also typically include a detailed verbal description of the actions being depicted. We chose to focus on these lifestyle vlogs because they are very popular, with tens of millions uploads on YouTube (Table 2.2). We also found that vlogs also capture a wide range of everyday activities; on average, we found thirty different visible human actions in five minutes of video.

In Chapter 4 we presented qualitative and quantitative analyses of the videos, which indicate that both the textual and visual information are rich sources for describing not only the actions, but why the actions in the videos are undertaken (action reasons).

Because of these characteristics, lifestyle vlogs are a rich data source for an in depth study of human actions and behaviours.

2. **Can machine learning models learn useful characteristics of human actions from lifestyle vlogs?**

   Action understanding algorithms evolve with the building of complex datasets.

   In Chapters 2, 3, 4 and 5 we showed that machine learning models can learn useful characteristic about human actions from lifestyle vlogs. We show that by building datasets based on the visual and textual information from lifestyle vlogs, and building models for complex tasks (action visibility classification, action localization, action reason identification and action co-occurrence identification), which are trained and tested on our datasets.

   We use the transcript of the videos to extract action names and align them with corresponding frames in the video. From the transcripts, the models can learn how humans can express the same action in different ways: "grab my Kindle", "do some reading", or "chill out". In Chapter 2 we showed that models can learn which of the actions mentioned in the transcript are visible in the video. This can lead to building large repositories of action and video pairs with minimal human intervention.

   In Chapter 3 we showed that using the textual, visual information and duration of an action, the models can learn where the action is localized in the video. The vlogs also contain overlapping actions or actions that tend to follow each other, which can provide models with rich information about our daily routines, that in turn can be used for the tasks of action prediction.

   In Chapter 4 we showed that models can use the transcripts and videos to learn why someone chooses to perform an action. This is due to how vloggers verbally express their intentions and feelings about the activities they perform.

   Most human actions are interconnected, as an action that ends is usually followed by the start of a related action and not a random one: after "waking up", one would "wash face" or "make breakfast" and not "sell books" or "go to bed". The interconnection of human actions is very well depicted in lifestyle vlogs, where vloggers visually record their everyday routine consisting of the activities they perform during a regular day. In Chapter 5 we used this information to address a task that can lead

to advances in action recognition systems: capturing the relations between human actions.

3. **Are multimodal models more effective than uni-modal models in solving the tasks and if so, how to combine different modalities?**

   In Chapters 2, 3, 4, 5 we presented multimodal models for solving the tasks of human action visibility classification, human action localization, human action reason identification and human action co-occurrence identification. We also compared the multimodal models with multiple uni-modal baselines. In Chapter 2, we showed results obtained using the multimodal model for different sets of input features: concreteness, context, POS. The model that uses all the input features available leads to the best results, improving significantly over the text-only and video-only methods. The visual features are concatenated and run through an LSTM [73]. The output is then concatenated with the textual features and the addition information and run through a three-layer feed forward network with dropout.

   The results for the task of human action localization were shown in Chapter 3. In order to combine the information from both textual and visual modalities, we used the MPU [14] model. The MPU model is composed of vector element-wise addition, vector element-wise multiplication and vector concatenation followed by a Fully Connected layer. The outputs from all three operations are concatenated to construct a multimodal representation. The resulting representation is given as input to a linear layer and finally to a sigmoid function to obtain a similarity score.

   For the task of human action reason identification (Chapter 4), we found that the textual model performs slightly better than the multimodal model. We believe that a data imbalanced split (more reasons are inferred from text than from videos) might be a reason for why the multimodal model does not perform as well as the text model. We combine the information from both textual and visual modalities, by using a T5 [42] model. We input the visual features to a T5 encoder after the transcript text tokens. The text input is passed through an embedding layer, while the video features are passed through a linear layer.

   The task of human action co-occurrence identification (Chapter 5) was represented as a link prediction task: the actions are represented as nodes in a graph and the co-occurrence relation between two actions is represented through a link between the actions. We ran experiments with various types of data representations: textual, visual and graph, using the graph topology. The learning-based model, an SVM [43] classifier, using the concatenation of all input representations (textual, visual, graph)

and all graph heuristic scores obtains the highest accuracy score, improving significantly over the text-only, visual-only and graph-only representations.

Experiments from Chapters 2, 3, 4, 5 revealed that both textual and visual modalities contribute to solving the tasks.

4. **Can we build automatic models for solving physical tasks related to human action understanding such as action visibility classification and action localization?**

   In Chapters 2 and 3, we presented the lifestyle vlog datasets we built for the tasks of human action visibility classification and action localization.

   For the task of human action visibility classification (Chapter 2), we built a multimodal neural architecture that combines encoders for the video and text modalities, as well as additional information (e.g., concreteness). We tested our models on our lifestyle vlogs dataset. The model that uses all the input features available leads to the best results, improving significantly over the text-only and video-only methods.

   For the task of human action localization (Chapter 3), we proposed a two-stage method which we called 2SEAL (2-StagE Action Localization). We found that shorter actions can be localized mainly based on the temporal information inferred from the transcript, whereas longer actions are often temporally shifted with respect to their mention in the transcript and thus can benefit from a multimodal model. We thus devised an architecture that first aims to predict whether the action is short or long, and correspondingly activates a transcript alignment (for short actions) or a multimodal model (for long actions).

   The results demonstrated that the tasks are challenging and there is room for improvement for future work models.

5. **Can we build automatic models for solving commonsense tasks related to human action understanding such as action reason classification and action co-occurrence identification?**

   Empirical experiments on the datasets presented in Chapters 4 and 5 demonstrated the effectiveness of our proposed method in comparison to several competitive baselines.

   For the task of human action reason classification (Chapter 4), we built a multimodal Fill-in-the-blanks model. The span of text "because _____" is introduced in the video

transcript, after the appearance of the action. We trained a language model to compute the likelihood of filling in the blank with each of the candidate reasons. For this purpose, we used T5 [42], an encoder-decoder transformer [173] pre-trained model, to fill in blanks with text.

For the task of human action co-occurrence identification (Chapter 5), the task can be represented as a graph link prediction task. We explore different models with different input representations (textual, visual and graph). We group the models as described in the related work link prediction section: heuristic-based models (graph topology models [181, 182, 183, 199, 184, 185, 186, 179]), embedding-based models (cosine similarity and graph neural networks [188, 201, 202, 189]), and learning-based models (SVM [43] models).

The results demonstrated that the tasks are challenging and there is room for improvement for future work models.

6.2 The Way Onward

This thesis has shed light on several research questions concerned with human action understanding, and at the same time it has opened a number of avenues for future research in multimodal representations for action understanding.

We have shown that multimodal systems generally outperform systems that rely on one modality at a time. However, there is still a long way ahead for building machines that can "watch" a lifestyle vlog and fully understand what the humans say and do while performing actions.

One direction for future research is to extend these datasets and use them to train more complex models for human action understanding. Other sources of data, like audio or video viewers comments can be useful for learning more about human actions and behaviours. The audio can provide more information about which activities are depicted in the video, while the comments can provide information about human values and behaviours.

A second potential line of research is concerned with building models that combine the multiple modalities more effectively. There are multiple ways of combining textual and visual information and we provided only a few approaches. More structured information from knowledge bases like Atomic [7], VerbNet [204] or ConceptNet [6] can enhance the models with commonsense knowledge.

Another direction for future research is building multi-task models for human action understanding. In our research, we build a separate model for each human action understanding

task. However, having only one model that can complete multiple tasks would be much more efficient and applicable in real life. A model that can learn to both recognize, localize and predict human actions would be more efficient and robust as it uses information from multiple sources.

Finally, we hope that the resources, tools, and methodologies presented in this thesis encourage further research on building multimodal models for action understanding from lifestyle vlogs.

BIBLIOGRAPHY

[1] Sarah R Fletcher and Philip Webb. Industrial robot ethics: The challenges of closer human collaboration in future manufacturing systems. In *A world with robots*, pages 159–169. Springer, 2017.

[2] Jamshed Iqbal, Zeashan Hameed Khan, and Azfar Khalid. Prospects of robotics in food industry. *Food Science and Technology International*, 37:159–165, 2017.

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017.

[4] David F. Fouhey, Weicheng Kuo, Alexei A. Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4991–5000, 2018.

[5] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020.

[6] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972.

[7] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *AAAI*, pages 3027–3035, 2019. URL https://doi.org/10.1609/aaai.v33i01.33013027.

[8] Ryan L. Boyd, Steven R. Wilson, James W. Pennebaker, Michal Kosinski, David Stillwell, and Rada Mihalcea. Values in words: Using language to evaluate and understand personal values. In *ICWSM*, 2015.

[9] Steven Wilson and Rada Mihalcea. Predicting human activities from user-generated content. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2572–2582, 2019.

[10] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3d: Generic features for video analysis. *ArXiv*, abs/1412.0767, 2014.

[11] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] Da Zhang, Xiyang Dai, Xin Wang, and Yuan fang Wang. S3d: Single shot multi-span detector via fully 3d convolutional networks. *ArXiv*, abs/1807.08069, 2018.

[13] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7745–7754, 2019.

[14] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5267–5275, 2017.

[15] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks formoment localization with natural language. In *AAAI*, 2020.

[16] K. Soomro, A. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, abs/1212.0402, 2012.

[17] Seyed Morteza Safdarnejad, Xiaoming Liu, Lalita Udpa, Brooks Andrus, John Wood, and Dean Craven. Sports videos in the wild (svw): A video dataset for sports analysis. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 1:1–7, 2015.

[18] Marcus Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1194–1201, 2012.

[19] Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *arXiv preprint arXiv:1503.01558*, 2015.

[20] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

[21] Luowei Zhou, Chenliang Xu, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, pages 7590–7598, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17344.

[22] Antoine Miech, D. Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, I. Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2630–2640, 2019.

[23] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, L. Zhao, Jiwen Lu, and J. Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1207–1216, 2019.

[24] Amirsina Torfi, Rouzbeh A. Shirvani, Yaser Keneshloo, Nader Tavvaf, and Edward A. Fox. Natural language processing advancements by deep learning: A survey. *ArXiv*, abs/2003.01200, 2020.

[25] Hongzhi Yin, Bin Cui, Ling Chen, Zhiting Hu, and Zi Huang. A temporal context-aware model for user behavior modeling in social media systems. *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, 2014.

[26] Ethan Fast, William McGrath, Pranav Rajpurkar, and Michael S. Bernstein. Augur: Mining human behaviors from fiction to power interactive systems. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016.

[27] Steven R. Wilson and Rada Mihalcea. Measuring semantic relations between human activities. In *IJCNLP*, 2017.

[28] Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6384–6392, May 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16792.

[29] Karin Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania, 2005.

[30] Michalis Vrigkas, Christophoros Nikou, and I. Kakadiaris. A review of human activity recognition methods. *Frontiers Robotics AI*, 2:28, 2015.

[31] Dan Roth. Learning to resolve natural language ambiguities: A unified approach. In *AAAI/IAAI*, 1998.

[32] Christopher D. Manning. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *CICLing*, 2011.

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

[34] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, A. Natsev, G. Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *ArXiv*, abs/1609.08675, 2016.

[35] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *arXiv preprint arXiv:2106.02636*, 2021.

[36] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5296–5305, 2017.

[37] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[38] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision (ECCV)*, pages 510–526. Springer, 2016.

[39] Wei Shi and Vera Demberg. Next sentence prediction helps implicit discourse relation classification within and across domains. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5790–5796, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1586. URL https://aclanthology.org/D19-1586.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[41] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

[42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[43] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[44] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[45] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6047–6056, 2018.

[46] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, 2015.

[47] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 588–595, 2014.

[48] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2117–2126, 2017.

[49] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[50] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision (ECCV)*, pages 20–36. Springer, 2016.

[51] Sreemanananth Sadanand and Jason J Corso. Action bank: A high-level representation of activity in video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1234–1241. IEEE, 2012.

[52] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, 2016.

[53] Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L Berg, and Dimitris Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 28–35. IEEE, 2012.

[54] Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*, 2017.

[55] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will

Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.

[56] Christoph Bregler. Learning and recognizing human dynamics in video sequences. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 568–574. IEEE, 1997.

[57] Mehrnaz Fani, Helmut Neher, David A Clausi, Alexander Wong, and John Zelek. Hockey action recognition via integrated stacked hourglass network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 29–37, 2017.

[58] Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial intelligence review*, 43(1):1–54, 2015.

[59] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

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

[61] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[62] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and vision computing*, 60:4–21, 2017.

[63] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[64] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[65] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail. In *German conference on pattern recognition*, pages 184–195. Springer, 2014.

[66] Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse

object stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2634–2641, 2013.

[67] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2758–2766, 2017.

[68] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017.

[69] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European Conference on Computer Vision (ECCV)*, pages 505–520. Springer, 2014.

[70] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5179–5187, 2015.

[71] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015.

[72] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision (ECCV)*, pages 817–834. Springer, 2016.

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

[74] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[75] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[76] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. Tvqa: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1379, 2018.

[77] Xun Xu, Timothy Hospedales, and Shaogang Gong. Semantic embedding space for zero-shot action recognition. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 63–67. IEEE, 2015.

[78] Qiuxia Wu, Zhiyong Wang, Feiqi Deng, Zheru Chi, and David Dagan Feng. Realistic human action recognition with multimodal feature selection and fusion. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4):875–885, 2013.

[79] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750, 2014.

[80] Joseph L Fleiss and Jacob Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619, 1973.

[81] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

[82] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[83] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911, 2014.

[84] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7263–7271, 2017.

[85] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[86] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009.

[87] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[88] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[89] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[90] Oana Ignat, Laura Burdick, Jia Deng, and Rada Mihalcea. Identifying visible actions in lifestyle vlogs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6406–6417, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1643. URL https://aclanthology.org/P19-1643.

[91] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. *CVPR 2011*, pages 3169–3176, 2011.

[92] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 988–996, 2017.

[93] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[94] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. *arXiv preprint arXiv:1809.01337*, 2018.

[95] Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. Mac: Mining activity concepts for language-based temporal localization. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 245–253. IEEE, 2019.

[96] Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander Hauptmann. Excl: Extractive clip localization using natural language descriptions. *arXiv preprint arXiv:1904.02755*, 2019.

[97] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3537–3545, 2019.

[98] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. *arXiv preprint arXiv:1912.06430*, 2019.

[99] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola. Stacked attention networks for image question answering. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–29, 2015.

[100] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[101] Jie Lei, Licheng Yu, Tamara L. Berg, and Mohit Bansal. Tvqa+: Spatio-temporal grounding for video question answering. *ArXiv*, abs/1904.11574, 2019.

[102] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 19–27. ACM, 2018.

[103] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR '07*, 2007.

[104] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[105] Shruti Palaskar, Jindrich Libovickỳ, Spandana Gella, and Florian Metze. Multimodal abstractive summarization for how2 videos. *arXiv preprint arXiv:1906.07901*, 2019.

[106] Bryan A Plummer, Matthew Brown, and Svetlana Lazebnik. Enhancing video summarization via vision-language embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5781–5789, 2017.

[107] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019.

[108] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013. doi: 10.1162/tacl_a_00207. URL https://aclanthology.org/Q13-1003.

[109] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. Temporal localization of moments in video collections with natural language. *arXiv preprint arXiv:1907.12763*, 2019.

[110] Mingzhe Wang, Mahmoud Azab, Noriyuki Kojima, Rada Mihalcea, and Jia Deng. Structured matching for phrase localization. In *European Conference on Computer Vision*, pages 696–711. Springer, 2016.

[111] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–216, 2018.

[112] Botian Shi, Lei Ji, Yaobo Liang, Nan Duan, Peng Chen, Zhendong Niu, and Ming Zhou. Dense procedure captioning in narrated instructional videos. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 6382–6391, 2019.

[113] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017.

[114] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

[115] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *ACL*, 2014.

[116] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812, 2017.

[117] Yitian Yuan, Tao Mei, and Wenwu Zhu. To find where you talk: Temporal sentence localization in video with attention based location regression. *ArXiv*, abs/1804.07014, 2018.

[118] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. Attentive moment retrieval in videos. In *SIGIR '18*, 2018.

[119] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

[120] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7464–7473, 2019.

[121] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019.

[122] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[123] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6325–6334, 2016.

[124] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*, 2018.

[125] Drew A Hudson and Christopher D Manning. Gqa: a new dataset for compositional question answering over real-world images. *arXiv preprint arXiv:1902.09506*, 2019.

[126] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *arXiv preprint arXiv:1804.09626*, 2018.

[127] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016.

[128] Klaus Krippendorff. Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement*, 30(1):61–70, 1970.

[129] Tomas Mikolov, Kai Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[130] Alex Graves. Supervised sequence labelling with recurrent neural networks. In *Studies in Computational Intelligence*, 2008.

[131] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2015.

[132] Tanvi S Motwani and Raymond J Mooney. Improving video activity recognition using object recognition and text mining. In *ECAI*, volume 1, page 2, 2012.

[133] Gunnar A Sigurdsson, Olga Russakovsky, and Abhinav Gupta. What actions are needed for understanding human actions in videos? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2137–2146, 2017.

[134] Zheng Shou, J. Chan, Alireza Zareian, K. Miyazawa, and S. Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1417–1426, 2017.

[135] Du Tran, Heng Wang, L. Torresani, Jamie Ray, Y. LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[136] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A. Ross, Jia Deng, and R. Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018.

[137] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253, 2019.

[138] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210, 2019.

[139] A. Karpathy, G. Toderici, Sanketh Shetty, Thomas Leung, R. Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[140] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, 2020.

[141] Gunnar A. Sigurdsson, Olga Russakovsky, and A. Gupta. What actions are needed for understanding human actions in videos? *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2156–2165, 2017.

[142] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *ArXiv*, abs/1806.11230, 2018.

[143] Henry L. Tosi. A theory of goal setting and task performance. *Academy of Management Review*, 16:480–483, 1991.

[144] Thomas Gilovich, Dale Griffin, and Daniel Kahneman. *Heuristics and biases: The psychology of intuitive judgment*. Cambridge university press, 2002.

[145] Carl Vondrick, Deniz Oktay, H. Pirsiavash, and A. Torralba. Predicting motivations of actions by leveraging text. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2997–3005, 2016.

[146] Jinyoung Yeo, Gyeongbok Lee, Gengyu Wang, Seungtaek Choi, Hyunsouk Cho, Reinald Kim Amplayo, and Seung-won Hwang. Visual choice of plausible alternatives: An evaluation of image-based commonsense causal reasoning. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1316.

[147] Hongming Zhang, Yintong Huo, Xinran Zhao, Yangqiu Song, and Dan Roth. Learning contextual causality between daily events from time-consecutive images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1752–1755, June 2021.

[148] Zhiyuan Fang, Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. Video2Commonsense: Generating commonsense descriptions to enrich video captioning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 840–860, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.61. URL https://www.aclweb.org/anthology/2020.emnlp-main.61.

[149] Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1044. URL https://aclanthology.org/P17-1044.

[150] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

[151] Sandro Pezzelle, Claudio Greco, Greta Gandolfi, Eleonora Gualdoni, and Raffaella Bernardi. Be Different to Be Better! A Benchmark to Leverage the Complementarity of Language and Vision. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2751–2767, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.248. URL https://aclanthology.org/2020.findings-emnlp.248.

[152] Stuart Synakowski, Q. Feng, and A. Martínez. Adding knowledge to unsupervised algorithms for the recognition of intent. *International Journal of Computer Vision*, pages 1–18, 2020.

[153] Menglin Jia, Zuxuan Wu, Austin Reiter, Claire Cardie, Serge Belongie, and Ser-Nam Lim. Intentonomy: a dataset and study towards human intent understanding. In *2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[154] Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. GLUCOSE: GeneraLized and COntextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.370. URL https://aclanthology.org/2020.emnlp-main.370.

[155] Jae Sung Park, Chandra Bhagavatula, Roozbeh Mottaghi, Ali Farhadi, and Yejin Choi. Visualcomet: Reasoning about the dynamic context of a still image. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 508–524, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58558-7.

[156] Dandan Song, Siyi Ma, Zhanchen Sun, Sicheng Yang, and Lejian Liao. Kvl-bert: Knowledge enhanced visual-and-linguistic bert for visual commonsense reasoning.

Knowledge-Based Systems, 230:107408, 2021. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2021.107408. URL https://www.sciencedirect.com/science/article/pii/S0950705121006705.

[157] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[158] J. Xu, Tao Mei, Ting Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016.

[159] Tom Michael Mitchell, William W. Cohen, Estevam R. Hruschka, Partha P. Talukdar, Bo Yang, J. Betteridge, Andrew Carlson, B. D. Mishra, Matt Gardner, Bryan Kisiel, J. Krishnamurthy, N. Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapandula Nakashole, Emmanouil Antonios Platanios, Alan Ritter, M. Samadi, Burr Settles, R. C. Wang, D. Wijaya, A. Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. Never-ending learning. *Communications of the ACM*, 61:103 – 115, 2015.

[160] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. NEIL: Extracting visual knowledge from web data. *2013 IEEE International Conference on Computer Vision*, 2013.

[161] F. Murtagh and P. Legendre. Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, 31:274–295, 2014.

[162] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL https://doi.org/10.5281/zenodo.1212303.

[163] Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1191. URL https://aclanthology.org/D18-1191.

[164] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1034. URL https://aclanthology.org/P15-1034.

[165] J. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382, 1971.

[166] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410.

[167] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[168] Vladimir Iashin and Esa Rahtu. A better use of audio-visual cues: Dense video captioning with bi-modal transformer. In *British Machine Vision Conference (BMVC)*, 2020.

[169] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017. URL https://arxiv.org/abs/1609.09430.

[170] Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1404. URL https://aclanthology.org/D19-1404.

[171] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

[172] Santiago Castro, Ruoyao Wang, Pingxuan Huang, Ian Stewart, Oana Ignat, Nan Liu, Jonathan C. Stroud, and Rada Mihalcea. FIBER: Fill-in-the-blanks as a challenging video understanding evaluation framework. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2925–2940, Dublin, Ireland, May 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.acl-long.209/.

[173] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach,

R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[174] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[175] Oana Ignat, Santiago Castro, Hanwen Miao, Weijia Li, and Rada Mihalcea. Whyact: Identifying action reasons in lifestyle vlogs. In *EMNLP*, 2021.

[176] Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In So Kweon. Detecting human-object interactions with action co-occurrence priors. In *ECCV*, 2020.

[177] Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In-So Kweon. Acp++: Action co-occurrence priors for human-object interaction detection. *IEEE Transactions on Image Processing*, 30:9150–9163, 2021.

[178] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016.

[179] David Liben-Nowell and Jon M. Kleinberg. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, 58: 1019–1031, 2007.

[180] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A-statistical Mechanics and Its Applications*, 553:124289, 2020.

[181] Mark E. J. Newman. Clustering and preferential attachment in growing networks. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 64 2 Pt 2:025102, 2001.

[182] Paul Jaccard. Etude de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37:547–579, 01 1901. doi: 10.5169/seals-266450.

[183] Gerard Salton and Michael J McGill. *Introduction to modern information retrieval*. mcgraw-hill, 1983.

[184] Lada A. Adamic and Eytan Adar. Friends and neighbors on the web. *Soc. Networks*, 25:211–230, 2003.

[185] Erzsébet Ravasz, Audrey Somera, D A Mongru, Zoltán N. Oltvai, and A.-L. Barabasi. Hierarchical organization of modularity in metabolic networks. *Science*, 297:1551 – 1555, 2002.

[186] Tao Zhou, Linyuan Lü, and Yicheng Zhang. Predicting missing links via local information. *The European Physical Journal B*, 71:623–630, 2009.

[187] Hisashi Kashima and Naoki Abe. A parameterized probabilistic model of network evolution for supervised link prediction. *Sixth International Conference on Data Mining (ICDM'06)*, pages 340–349, 2006.

[188] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[189] Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *ArXiv*, abs/1609.02907, 2017.

[190] Mark Yatskar, Luke Zettlemoyer, and Ali Farhadi. Situation recognition: Visual semantic role labeling for image understanding. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[191] Matteo Ruggero Ronchi and Pietro Perona. Describing common human visual actions in images. *arXiv preprint arXiv:1506.02203*, 2015.

[192] Beth Levin. *English verb classes and alternations: A preliminary investigation*. University of Chicago press, 1993.

[193] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[194] Tadeusz Caliński and Joachim Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3:1–27, 1974.

[195] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:224–227, 1979.

[196] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[197] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[198] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *ArXiv*, abs/2104.08860, 2021.

[199] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

[200] Boyao Zhu and Yongxiang Xia. Link prediction in weighted networks: A weighted mutual information model. *PLoS ONE*, 11, 2016.

[201] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Attributed network embedding via subspace discovery. *Data Mining and Knowledge Discovery*, 33: 1953–1980, 2019.

[202] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, 2017.

[203] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information and lexicography. In *ACL*, 1989.

[204] Karin Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania, 2005.