Learning Human Action Representations from Temporal Context in Lifestyle Vlogs

Oana Ignat Santiago Castro Weiji Li Rada Mihalcea

University of Michigan - Ann Arbor, USA

`oignat@umich.edu`

Abstract

We address the task of human action representation and show how the approach of generating word representations based on co-occurrence can be adapted to generate human action representations by analyzing their co-occurrence in videos. To this end, we formalize the new task of human action co-occurrence identification in online videos, i.e., determine whether two human actions are likely to co-occur in the same interval of time. We create and make publicly available the CO-ACT (Action Co-occurrence) dataset, consisting of a large graph of ∼12k co-occurring pairs of visual actions and their corresponding video clips. We describe graph link prediction models that leverage visual and textual information to automatically infer if two actions are co-occurring. We show that graphs are particularly well suited to capture relations between human actions, and the learned graph representations are effective for our task and capture novel and relevant information across different data domains.

1 Introduction

Action understanding is a long-standing goal in the development of intelligent systems that can meaningfully interact with humans, with recent progress made in several fields including natural language processing (Fast et al., 2016; Wilson and Mihalcea, 2017, 2019), computer vision (Carreira and Zisserman, 2017; Shou et al., 2017; Tran et al., 2018; Chao et al., 2018; Girdhar et al., 2019; Feichtenhofer et al., 2019), data mining (Kato et al., 2018; Xu et al., 2019), and others. Many of the action understanding systems developed to date, however, rely mostly on pattern memorization and do not effectively understand the action, which makes them fragile and unable to adapt to new settings (Sigurdsson et al., 2017; Kong and Fu, 2018).

Effective action understanding requires reliable action representations. In this paper, we introduce a strategy to generate contextual representations for human actions by adopting an approach for creating
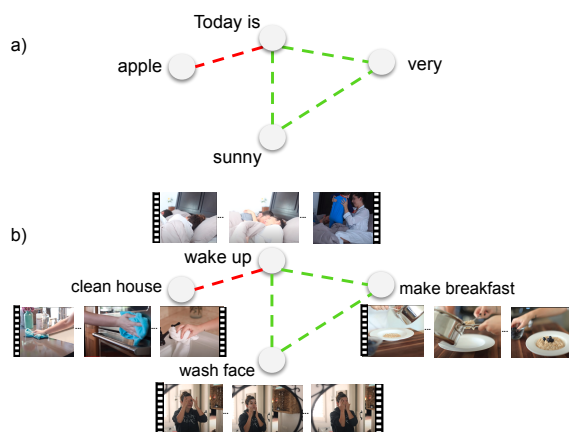

Figure 1: We draw inspiration from contextual word representations to create novel action representations based on video temporal context. Specifically, when predicting the next word in a sentence, it is more expected to see certain words, for instance, after "Today is" an expected word is "sunny" and not "apple". Similarly, human actions also follow a certain pattern, for instance, after "waking up", an expected next action is to "wash the face" and not to "clean the house".

word representations based on co-occurrence information. In linguistics, co-occurrence is defined as an above-chance frequency of ordered occurrence of two adjacent terms in a text corpus. For example, if the concepts "peanut butter", "jelly", and "sandwich" appear more often together than apart, they would be grouped into a concept co-occurrence rule. Co-occurrence is a building block concept for word representations and language models. We adapt this approach to human actions, which also have their own co-occurrence relations, expressed as temporal context. 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 break-fast" co-occur in a short interval of time, while "wake up", "clean house" or "go to bed" do not. A natural way to model the connections between human actions is through a graph representation, where actions are represented as nodes, and their co-occurrences are represented as edges (Fig. 1).

The interconnection of human actions is well captured in lifestyle vlogs, where vloggers visually record their everyday routine consisting of the activities they perform during a regular day (Fouhey et al., 2018; Ignat et al., 2019, 2021). We collect a dataset of lifestyle vlogs from YouTube that are currently very challenging for systems to solve.

Contributions. Our paper makes four main contributions. First, we show how the approach to generating word representations based on co-occurrence can be adapted to generating representations for human actions by analyzing their co-occurrence in videos. To this end, **we formalize the new human action co-occurrence identification task** in online videos. Second, **we introduce a new dataset**, CO-ACT, 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, multi-modal, and graph-based action representations. Finally, **we show that our action representations based on co-occurrence capture novel and relevant information across different data domains**, which leads to rich avenues for future work for improving action representation and making progress toward the broader goal of action understanding.

2 Related Work

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

Human Action Co-occurrence. Recent work shows that action co-occurrence priors (Kim et al., 2020, 2021) increase the performance of human-object interaction models and lead to more effective training, especially in long-tail classes. Unlike our work, they assume that the action co-occurrence information is provided and 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. (2016) 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, 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 (Liben-Nowell and Kleinberg, 2007).

Link prediction approaches can be categorized into three main categories (Kumar et al., 2020): similarity-based/heuristic (Newman, 2001; Jaccard, 1901; Salton and McGill, 1983; Adamic and Adar, 2003; Ravasz et al., 2002; Zhou et al., 2009; Liben-Nowell and Kleinberg, 2007); probabilistic-based (Kashima and Abe, 2006); and dimensionality reduction-based (e.g., embedding-based or other learning approaches; Grover and Leskovec, 2016; Kipf and Welling, 2017).

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.

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 on instructional videos (Miech et al., 2019) and lifestyle vlogs (Fouhey et al., 2018; Ignat et al., 2019, 2021, 2022) are other examples of web supervision. The latter is similar to our work as they analyze online vlogs, but unlike ours, their focus is on action detection or the reasons behind actions and not on action co-occurrence.

3 Dataset

To develop and test models for determining if two actions co-occur, we compile a novel dataset, which we refer to as CO-ACT (Action Co-

occurrencE).

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, etc. 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 data analyses in Appendix A.2.

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').

Therefore, we first ensure 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 (Yatskar et al., 2016), COCO-a (Ronchi and Perona, 2015) and Levin (Levin, 1993).[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 (Honnibal et al., 2020) 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 a 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 (Honnibal et al., 2020) 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.

Quality Assurance. As described above, we perform multiple steps to ensure the actions appear in the videos. First, we manually select 20 YouTube channels from vloggers with high-quality filming styles, who usually provide detailed visual and textual descriptions of their actions. Second, we automatically extract actions that contain visual verbs. We manually check around 100 extracted actions to see if they are parsed well and if they correctly match their corresponding video and transcript context. Third, we automatically filter out videos that do not contain any transcripts or no significant motion. We filter out the motionless videos by following the procedure from Ignat et al. (2019): we sample one out of every one hundred frames of the videos and compute the 2D correlation coefficient between these sampled frames. We filter out all the videos with a median of the values greater than a threshold (0.8). We manually check around 100 (action, video) pairs to see if they correctly match and find around 18 complete misalignments. Finally, to mediate the misalignment and obtain diverse filming perspectives, we randomly sample up

---

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

to 10 video clips for each action, which increases the chances that the action is present in at least one video. Random examples of actions and their video-frames are found in sample-frames.

3.2 Data Pre-processing

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

Action Co-occurrence Selection. 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 less than 10 seconds away from each other. The 10 seconds is an *intermediate value threshold* we set after experimenting with other values. This threshold controls the scale of time we choose to focus on when collecting co-occurring actions: e.g., mostly short actions (e.g., "open fridge", "get milk") are captured in a small interval of time (1-5 sec), while longer intervals allow for longer and more diverse actions to co-occur (e.g., "prepare meal"). We choose an intermediate value that allows for both shorter and longer actions to co-occur[3]. Our intuition is that modeling the relations between both shorter and longer actions would result in learning more comprehensive information about human actions. We also consider the in-depth analysis of this threshold and its downstream effects as an interesting future work direction and our framework allows for effortless threshold tune.

For computing the distance in time between two actions, we use the transcript time stamps. This allows scaling data with no constraints from the annotation budget. The transcript time stamps do not always match the time the action appears in the video. However, this hardly impacts our task because the actions mentioned in the transcript usually follow the order from the video. Furthermore, we mediate misalignments by collecting multiple videos per action and filtering steps described in the previous section.

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". To avoid such repetitions, we group similar actions by clustering all actions. We represent each action using the pre-trained model Sentence-BERT (Reimers and Gurevych, 2019) and apply Agglomerative Clustering (Murtagh and Legendre, 2014). We filter out the clusters of actions with less than two actions, as they are likely to be outliers 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" and "iron cloth" are renamed to "use iron".

We observe that the clustering model is introducing some noise level as it does not perfectly cluster all actions. We tried to mitigate this with different Sentence-BERT pre-trained models for sentence similarity[4] and fine-tuning our clustering model hyper-parameters[5] based on automatic evaluation metrics for measuring the quality of clusters[6].

Action Graph Filtering. After we rename the actions based on clustering, we create a graph where the nodes represent the actions, and the 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 a weight 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 equals 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 1. More information (e.g., action frequency distributions, action pairs) can be found in Appendix A.

3.3 ACE vs. current Human Action Datasets

Many publicly available visual action datasets (Carreira and Zisserman, 2017; Soomro et al., 2012; Kuehne et al., 2011) do not have video transcripts and do not have videos with multiple actions presented in their natural order, therefore we cannot leverage the textual information and the relations

---

[3]The captured actions also depend on the filming style (e.g., vloggers could increase the filming time of normally short actions).

[4]sbert.net/docs/pretrained_models.html
[5]linkage distance threshold (1.5), linkage criterion (ward)
[6]Silhouette Coefficient (Rousseeuw, 1987), Calinski-Harabasz Score (Caliński and Harabasz, 1974), and Davies-Bouldin Index (Davies and Bouldin, 1979)

between actions, as we can do in our dataset.

The majority of human actions datasets with transcripts are restricted to one or few domains (e.g., cooking (Zhou et al., 2018) or instructional videos (Miech et al., 2019; Tang et al., 2019)). The main difference between lifestyle vlogs and instructional videos is the domain of the actions. Instructional videos are usually from just one domain (e.g., either cooking, repairing, or construction) and tend to have a specialized vocabulary (e.g., car repair). Lifestyle vlogs contain various everyday actions from multiple domains in the same video (cleaning, cooking, DIY, entertainment, personal care). Due to the diversity of domains in our data, our model learns not only the co-occurrence between in-domain actions (e.g., cooking: "cut potato" & "add onion") but also the relations from different domains (e.g., personal care and cooking: "wash face" & "make breakfast").

4 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 an edge exists between two given actions.[7]

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 (Reimers and Gurevych, 2019) calculated using the graph topology and the textual embeddings obtained from CLIP (Radford et al., 2021). 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 (Radford et al., 2021).

Video Representations. We use the CLIP model (Radford et al., 2021) 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 preprocess them as done before (Radford et al., 2021). We use the pre-trained Vision Transformer model ViT-B/16 (Dosovitskiy et al., 2021) to encode the

video frames and the textual information. We apply the model to each of the four frames and average their representations (Luo et al., 2021).

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 neighbor node embeddings are represented using either textual embeddings (Sentence-BERT; Reimers and Gurevych, 2019) or visual embeddings (CLIP; Radford et al., 2021). All the graph-based models described in the next section use graph topology information from the validation graph (see Section 5.1).

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

4.2 Action Co-occurrence Models

We explore many models with various 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 4.1, we run experiments with various types of data representations: Textual: Action and Action Transcript; Visual: Action, Video, and Multimodal (Action&Videos; the average between action and video visual embeddings); Graph: Action and Multi-modal (Action&Videos) using graph topology.

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%.

4.2.2 Heuristic-based Graph Topology Models

We apply several popular node similarity methods that only use graph topology information in the prediction process: Common Neighbours (Newman, 2001), Salton Index (Salton and McGill, 1983), Adamic-Adar Index (Adamic and Adar, 2003), Hub Promoted Index (Ravasz et al., 2002), and Shortest Path (Liben-Nowell and Kleinberg, 2007). Note that the heuristic-based methods do not use any data representations described in Section 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 set of nodes connected to node x and k_x be the degree of node x.

---

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

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

$$s_{xy} = |\Gamma(x) \cap \Gamma(y)| \tag{1}$$

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

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{k_x k_y}} \tag{2}$$

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

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\min\{k_x, k_y\}} \tag{3}$$

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.

$$s_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log k_z} \tag{4}$$

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

$$s_{xy} = \frac{1}{min\{l : path_{xy}^{<l>} exists\}} \tag{5}$$

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) to use the link weight information, as proposed in Zhu and Xia (2016). We find that using link weights achieves similar results as without them.

4.2.3 Embedding-based Models

Cosine Similarity. We compute the cosine similarity between their embeddings to determine if two given actions co-occur. 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 models (Kumar et al., 2020): attri2vec (Zhang et al., 2019), GraphSAGE (Hamilton et al., 2017), GCN (Kipf and Welling, 2017). GNN 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 GraphSAGE with 77.2% accuracy, while the best-performing topology model has an 82.9% accuracy (see Table 2). Therefore, we conclude that our task does not benefit from these neural models.

4.2.4 Learning-based Model

We run a support vector machine (SVM) (Cortes and Vapnik, 1995) classifier on each action pair to be classified as co-occurring. We concatenate all the input representations and the heuristic scores, and we standardize the features by removing the mean and scaling to unit variance. We fine-tune the model hyper-parameters (kernel, C, gamma) on the validation data using a grid search.

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.

5.1 Evaluation Data Split

We split the original graph into train, validation, and test graphs. 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 the train, validation, and test sets using a transductive split, which is considered the default evaluation splitting technique for link prediction models (Xu et al., 2018). Specifically, we randomly sample 10% of all existing edges from the original graph as 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 create the validation and the train data for the models. The validation graph is created by reducing the test graph, and the training graph is created by reducing the validation graph.

Model	Accuracy
BASELINE	
Random	50.0
HEURISTIC-BASED	
Common Neighbours	82.9
Salton Index	71.2
Hub Promoted Index	78.3
Adamic-Adar Index	82.9
Shortest Path	82.9
EMBEDDING-BASED	
Cosine similarity	82.8
attri2vec	65.7
GCN	77.2
GraphSAGE	78.1
LEARNING-BASED	
SVM	**91.1**

Table 2: Accuracy results for all the models.

5.2 Results and Ablations

Table 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 Table 2. Simple heuristics (common neighbors or shortest path) are enough to perform well.

Modality Ablation. The ablation results, split by input representation are shown in Table 3. We analyze how different input representations influence the model's 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 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 significantly better performance (80.9% and 91.1% accuracy, respectively). The visual and multi-modal embeddings are also valuable but perform worse than the textual embeddings. We hypothesize that CLIP embeddings might be affected by the time misalignment between the transcript and the video. However, the visual modality offers important information about human actions and can be used in future work with more robust visual models.

5.3 Downstream Task: Action Retrieval

Similar to how word embeddings have been used for word similarity and for retrieving similar words and documents (Mikolov et al., 2013; Devlin et al., 2019), our graph dataset enables *action similarity* and *similar action retrieval* leveraging action-specific properties in the multi-modal space.

To show the usefulness of our graph-based action embeddings, we test them on the *similar action retrieval* downstream task. Specifically, we compare two action representations: textual (Action Sentence-BERT embeddings) and graph-based (graph weighted average of neighbor nodes Action Sentence-BERT embeddings). In Fig. 2, we show the top three nearest neighbor actions from each of the representations for three random action queries from our dataset. We observe that *each representation captures different types of information*. The actions obtained with textual representations are more syntactically similar to the action query, sharing either the verb or the object. This can be undesirable, as many retrieved actions are too repetitive and not always relevant to the action query: e.g., "build desk": "build bookshelf", "build house". In contrast, the actions obtained with graph representations are more *diverse* and capture *location information*, i.e., actions expected to be temporally close in a video: e.g., "build desk": "use knife", "add storage", "put piece of wood".

Novelty vs. Relevance in Action Retrieval. A major focus in the field of Information Retrieval has been the development of retrieval models that maximize both the relevance and the novelty among higher-ranked documents (Carbonell and Goldstein-Stewart, 1998). For the task of action retrieval, we can approximate *relevance* through the *location* relevance of an action, and *novelty* through the *diversity* of the actions retrieved.

Diversity in Action Representations. Similar to word or document retrieval, diversity in action retrieval reflects novel results. To measure the *diversity* captured by the action representations, we compute the *overlap score* as the number of overlapping words between the action query and the retrieved top k action nearest neighbors, divided by the total number of words in the retrieved actions. For example, in Fig. 2, the action query "chop potato", for $k = 3$, the action kNNs using textual representations (in blue) have 3 overlapping words ("chop", "potato", "potato"), from a total

Model	INPUT REPRESENTATIONS						
	Textual		Visual			Graph	
	Action	Transcript	Action	Video	Action&Video	Action	Action&Video
Cosine 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 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.

k	INPUT REPRESENTATIONS	
	Textual	Graph
	DIVERSITY/ OVERLAP SCORE ↓	
3	0.35	**0.12**
5	0.31	**0.11**
10	0.26	**0.10**
Dataset	LOCATION / RECALL SCORE ↑	
Breakfast	0.16	**0.22**
COIN	0.23	**0.60**
EPIC-KITCHENS	0.14	**0.26**

Table 4: Scores measuring the difference of information, diversity, and location, between the action kNNs using different types of embeddings: textual and graph-based.

of 8 words, resulting in an overlap score of 3/8. We average the overlap scores across all the action queries in our dataset (2,262 unique actions) for $k \in 3, 5, 10$. Table 4 shows that the actions retrieved using our graph representations have around three times fewer overlapping words with the action query; i.e., they are more diverse than those retrieved using the textual representation.

Location in Action Representations. To quantify how much *location* information an action representation holds, we use three annotated action localization datasets: COIN (Tang et al., 2019), which includes instructional videos; EPIC-KITCHENS (Damen et al., 2018); and Breakfast (Kuehne et al., 2014, 2016). We use the training data to create an action co-occurrence graph and learn action graph representations and the testing data to test our action representations. For each action query in the test set, we obtain the actions localized before and after as the gold standard action neighbors. We also calculate the predicted action kNNs ($k = 3$) of the action query using textual and graph-based representations. To measure the location information, we compute the recall score between the gold standard action temporal neighbors and the predicted action kNNs. Table 4 shows that graph-based representations hold more location information than textual representations.
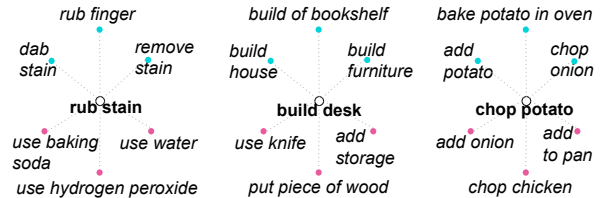

Figure 2: Top three action neighbors, obtained from textual (blue) and graph-based (purple) representations, for three random action queries from our dataset: "rub stain", "build desk", "chop potato".

Action representations that capture location information would likely benefit models in many computer vision applications, such as action localization, segmentation, or detection.[8] This leads to future research directions for effectively utilizing graph-based representations and co-occurring actions.

6 Conclusion

In this paper, we addressed the task of learning human action representations from co-occurring human actions in videos. We explored the genre of lifestyle vlogs and constructed CO-ACT, a new dataset of ∼12k pairs of visual actions and their corresponding clips. We evaluated models that leverage textual, visual, multi-modal, and graph information. We built CO-ACT and action co-occurrence identification models to capture human action relations, which leads to progress towards the goal of action understanding. We are the first to address this problem and to use graph representations in this setting. We showed that graph representations are useful for our task, capture information about human actions across diverse domains, and complement the representations learned from current language and visual models. The CO-ACT dataset and code are available at `https://github.com/MichiganNLP/vlog_action_co-occurrence`.

---

[8]For more on how our graph can be used in other downstream tasks, see Appendix A.1.

Ethics and Broad Impact Statement

Our dataset contains public YouTube vlogs, in which vloggers choose to share episodes of their daily life routine. We use the videos to detect co-occurring actions without relying on information about the person's identity, such as gender, age, or location.

The data can be used to better understand people's lives by looking at their daily routines and in which order they choose to perform their actions. The data contains videos of men and women and sometimes children, but most videos come from women. The routine videos present mostly ideal routines and are not comprehensive about all people's daily lives. Most of the people represented in the videos are middle-class Americans, and the language spoken is English.

In our data release, we only provide the YouTube URLs of the videos, so the creator of the videos can always have the option to remove them. YouTube videos are a frequent source of data in research papers (Miech et al., 2019; Fouhey et al., 2018; Abu-El-Haija et al., 2016), and we followed the typical process used by all this previous work of compiling the data through the official YouTube API and only sharing the URLs of the videos. We have the right to use our dataset how we use it, and we bear responsibility in case of a violation of rights or terms of service.

Limitations

Weak supervision from video transcripts. We use the weakly supervised time signal from automatically generated video transcripts without manual annotations. This allows for no limits in scale at the cost of some noise. To reduce the noise, we use multiple (up to 10) videos to obtain the temporal action information and perform various filtering steps described in the Quality Assurance subsection. Furthermore, the time information is used only to find the co-occurrence information between actions, not the actual time location of the actions; therefore, it is not necessary to be clear-cut.

Directed vs. Undirected graph representations. A directed graph also captures the order between the actions, which can be used in a future work direction for action prediction applications. However, an undirected graph is sufficient to obtain co-occurrence information, which suits our goal for our paper. We looked into transforming our graph into a directed one. However, we could not do this reliably because the transcripts do not pre-serve the exact order of the actions. This is due to how vloggers choose to verbally describe their routines: e.g. from "during washing my face, I will wake up" - it is not trivial to automatically extract the correct/natural order of the actions, as in this case, the result would be incorrect (wash face, then wake up). We tried modeling this using time keywords (e.g., "during", "after", "before") but due to the complexity of natural language, we found exceptions and other complex scenarios that could not be modeled automatically.

More advanced multimodal fusion techniques. More advanced multimodal fusion techniques might improve the performance, and we also include this direction in future work. However, in this paper, we focused on data collection and providing a set of comprehensive link prediction baselines: heuristic-based, embedding-based, and learning-based. These baselines are challenging, as demonstrated by the high accuracy results from Table 2.

Acknowledgements

References

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

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

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

Jaime G. Carbonell and Jade Goldstein-Stewart. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

João Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733.

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

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

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

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

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

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. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

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

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

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

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

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

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

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Oana Ignat, Laura Burdick, Jia Deng, and Rada Mihalcea. 2019. Identifying visible actions in lifestyle vlogs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6406–6417, Florence, Italy. Association for Computational Linguistics.

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

Oana Ignat, Santiago Castro, Yuhang Zhou, Jiajun Bao, Dandan Shan, and Rada Mihalcea. 2022. When did it happen? duration-informed temporal localization of narrated actions in vlogs. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 18:1 – 18.

Paul Jaccard. 1901. Etude de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37:547–579.

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

Keizo Kato, Yin Li, and Abhinav Kumar Gupta. 2018. Compositional learning for human object interaction. In *ECCV*.

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

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

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

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

H. Kuehne, A. B. Arslan, and T. Serre. 2014. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR)*.

H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. 2011. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Hilde Kuehne, Juergen Gall, and Thomas Serre. 2016. An end-to-end generative framework for video segmentation and recognition. In *Proc. IEEE Winter Applications of Computer Vision Conference (WACV 16)*, Lake Placid.

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

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

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

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

Antoine Miech, D. Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, I. Laptev, and Josef Sivic. 2019. 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.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

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

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

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

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. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

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

Nils Reimers and Iryna Gurevych. 2019. 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. Association for Computational Linguistics.

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

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

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

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

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

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

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

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

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

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

Steven R. Wilson and Rada Mihalcea. 2019. Predicting human activities from user-generated content. In *ACL*.

Bingjie Xu, Yongkang Wong, Junnan Li, Qi Zhao, and M. Kankanhalli. 2019. Learning to detect human-object interactions with knowledge. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2019–2028.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *ArXiv*, abs/1810.00826.

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

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

Luowei Zhou, Chenliang Xu, and Jason J. Corso. 2018. Towards automatic learning of procedures from web instructional videos. In *AAAI*.

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

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

A Appendix

A.1 How to use the graph in other datasets and downstream tasks

Our comprehensive graph can be extended to new data and new tasks. We intentionally included a large number of actions in our graph, making it comprehensive and exhaustive (see Table 1: 164 verbs/ 2,262 unique actions), precisely to increase the chance that actions from new data can be found in our graph.

We do not need to create a new graph for each new data or new task. Instead, we can directly use our provided learned action representations, which can also be fine-tuned on the new data. If there is a sufficiently similar (i.e., based on cosine similarity) class match between the actions we provide and the actions from the new data, then we can use our corresponding learned action representations. Otherwise, the new actions can be added to the graph, and new action representations can be computed. We provide code and guidelines on how to extend our graph and obtain new action representations.

Figure 3: Co-occurrence matrix for the top 50 most frequent actions in our dataset, CO-ACT. 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 matrix rows to highlight clusters. Best viewed in color.
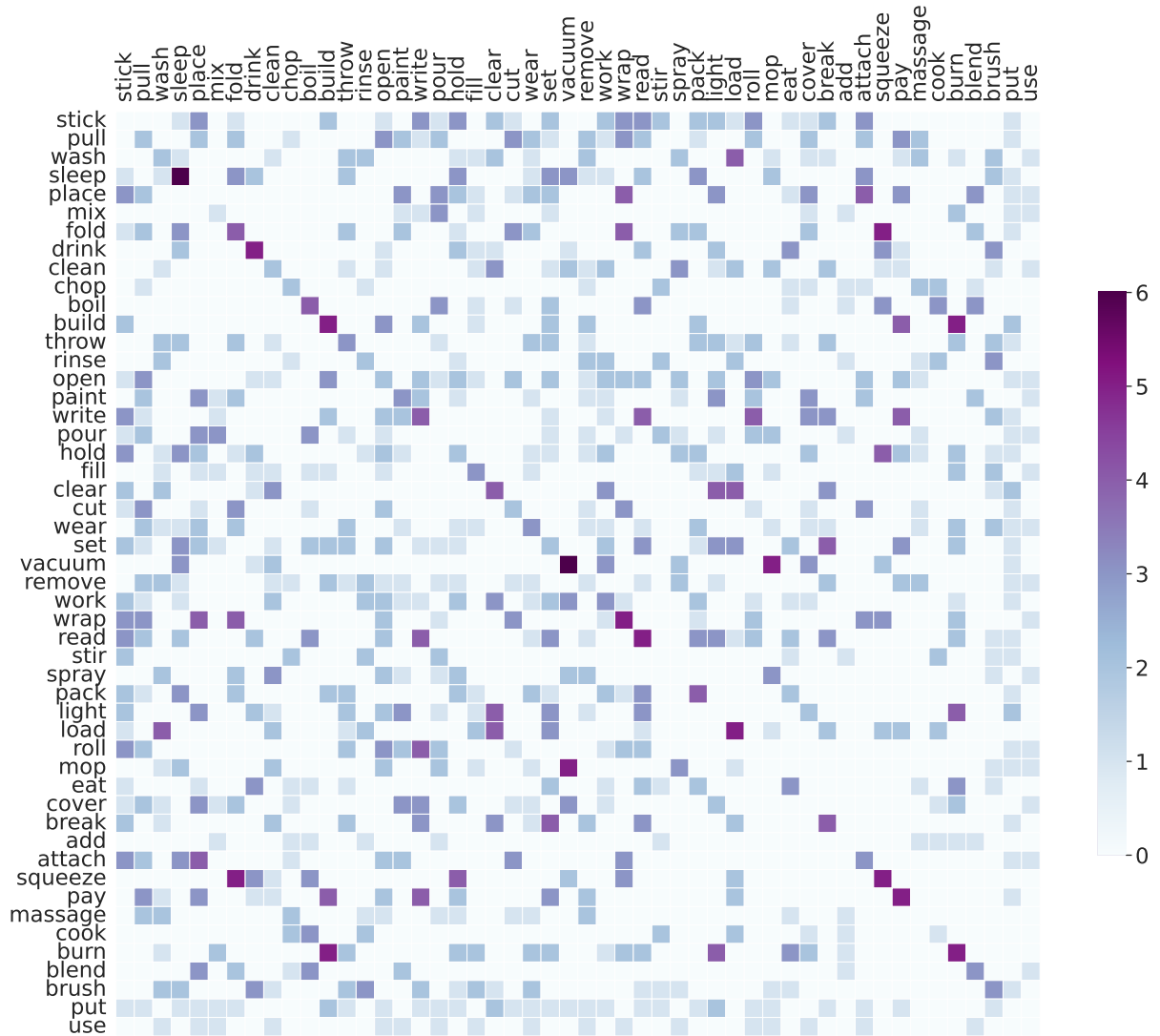
Figure 4: Co-occurrence matrix for the top 50 most frequent verbs in our dataset, CO-ACT. The scores are computed using the PPMI measure: actions with higher scores have a stronger co-occurrence relation and vice-versa. For better visualization, we sort the matrix rows to highlight clusters. Best viewed in color.

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
brush tooth, wash face	16		add, cut	256
curl eyelash, use mascara	15		clean, put	246
put in freezer, put in smoothie	15		eat, use	244
add tomato, cook on stove	15		eat, eat	201
clean bathtub, use broom	15		fill, use	191
...	...		...	...
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
use charcoal scrub, use sheep mask from store	2		fill, pump	2
use on drugstore, use product	2		build, draw	2
break surface of water, remove makeup	2		teach, work	2
brush hair, spray with hairspray	2		break, remove	2
fill brow, fill browser bed	2		brush, spray	2

Top 20 most and least frequent action pairs (left)
and verb pairs (right) in our dataset.

A.2 Data Analysis

We want to determine which actions co-occur the most in our dataset, as it may be valuable knowledge 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 actions known to co-occur with the previous action (e.g., given a previously recognized action "wake up", a likely next 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) (Church and Hanks, 1989). PMI is biased towards infrequent words, therefore we do not compute PMI for infrequent actions (that appear less than 10 times).

$$PPMI_{a_i, a_j} = \max(\log \frac{P_{a_i, a_j}}{P_{a_i} P_{a_j}}, 0) \qquad (6)$$

$$P_{a_i, a_j} = \frac{\#(a_i, a_j)}{\#action\ pairs}, P_{a_k} = \frac{\#a_k}{\#actions} \qquad (7)$$

Figure 5 shows the co-occurrence matrix for the top 20 most frequent actions. The most frequent actions are related to cooking. We can see how actions related to adding ingredients are co-occurring among 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"). Appendix A includes additional information: co-occurrence matrices of the top 50 most frequent actions and verbs (Figs. 3 and 4), top 20 actions and verb pairs co-occurring the most/least (Appendix A.1), actions and verbs distributions (Figs. 6 and 7), top 10 most frequent clusters (Fig. 8).

A.3 Action and Verb Distribution

A.4 Action Clustering

Recall that all the raw actions extracted from the transcript are clustered as described in Section 3.2. To analyze the content of the clusters, we show the 10 most frequent clusters using t-distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten and Hinton, 2008) (see Fig. 8)

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',
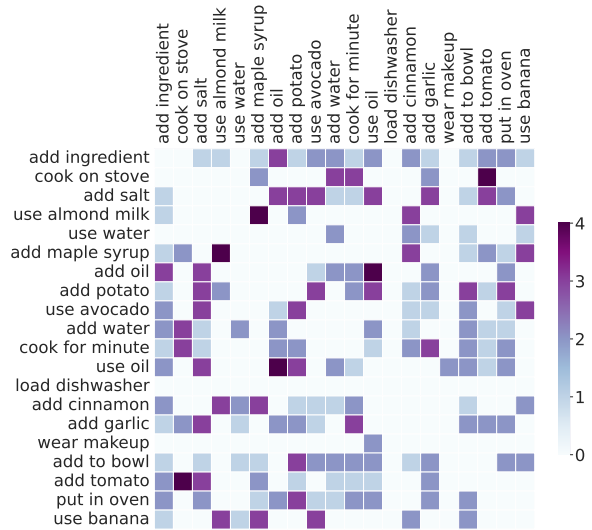

Figure 5: Co-occurrence matrix for the top 20 most frequent actions in our dataset, CO-ACT. The scores are computed using the PPMI measure: actions with higher scores have a stronger co-occurrence relation and vice-versa. For better visualization, we sort the matrix rows to highlight clusters. Best viewed in color.
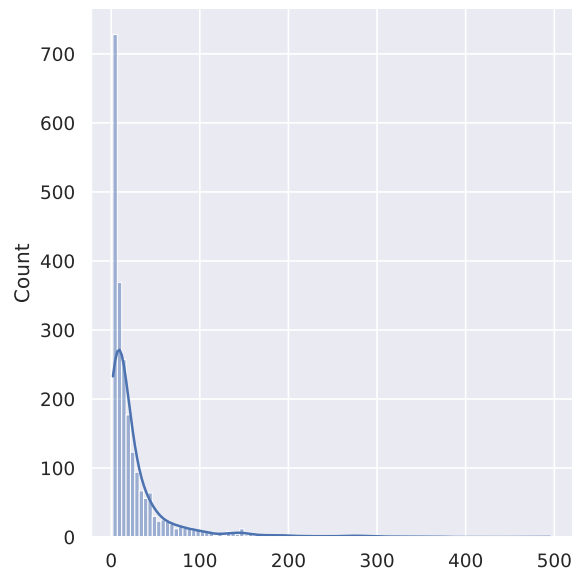

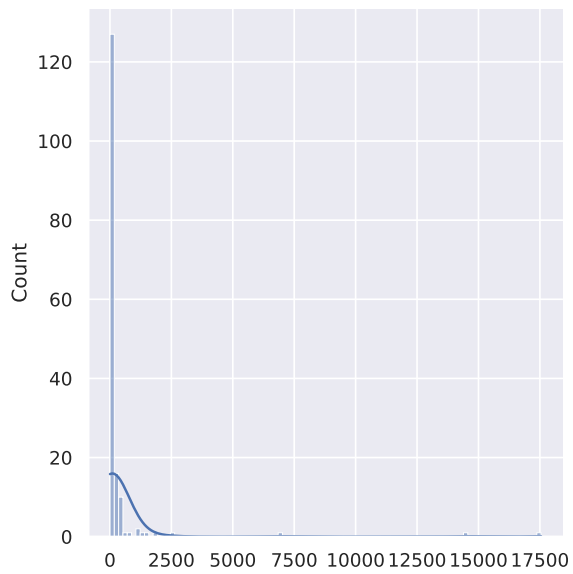Figure 6: Action distribution in our dataset, CO-ACT: count of actions frequencies.

16

Figure 7: Verb distribution in our dataset, CO-ACT: count of verb frequencies.

"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").

Figure 8: The t-SNE representation (Van der Maaten and Hinton, 2008) of the ten most frequent action clusters in our dataset. Each color represents a different action cluster. Best viewed in color.