ECE496 Proposal

Distilling Instructional Videos into Action Graphs

Project Number: 2020290

Supervisor: Animesh Garg

Administrator: Ross Gillett (Section 7)

Team Members: Mohsin Hasan, Md Nahian Rizve Khan, Sagar Patel

Date of Submission: October 22, 2020

Executive Summary

There is an abundance of instructional videos online that demonstrate how to perform tasks such as cooking spaghetti and meatballs. Building a computational understanding of videos using both the visual information and associated language in them is a research problem in the field of machine learning [1]. In the case of instructional videos, the problem of understanding the content is linked to the problem of determining the relations between the key instructional steps and the objects used in the task.

One common approach to this problem is to generate action graphs from videos which capture relations between objects over both space and time. Many algorithms producing action graphs focus on capturing low-level actions [2, 3], to be used for action recognition, but do not tackle the problem of finding relations among higher level instructions, which is what our project is interested in. The state-of-the-art solution for this more specific version of the problem [1] better captures the relations between key instructional steps. However, this model can be further improved in multiple ways, including by modifying the dataset used for training, the general architecture of the model, and the specific sub-components used in the model.

As such, our project proposes to produce an algorithm, which takes an instructional video with a time-aligned transcript as input, and outputs an action graph representing the video. The action graph is based on the graph in [1], and contains, as nodes, text descriptions of instructional steps and their objects, as well as bounding boxes around the locations of objects within a video frame corresponding to each step. “Reference” edges connect the objects to previous steps they are related to, and “grounding” edges connect objects to their visual bounding boxes.

The project will focus on design and implementation of this algorithm, with some subtasks such as object proposal and parsing, potentially being integrated from open-source implementations [4, 5]. The algorithm will use publicly available datasets to learn from and test performance on. Different existing datasets will be investigated for use since the choice of dataset has a large impact on the performance of learning algorithms. The performance of the algorithm will be evaluated using existing metrics for similar tasks. The project is required to take regular video input, and output graphs that are compatible with evaluation metrics and visually interpretable. The objectives for the project are based on maximizing accuracy while minimizing computational effort. Major milestones for this project include building a reference resolution model, a visual grounding model and an inference pipeline for video inputs.

Table of Contents

Background and Motivation

Background Research

With the rise of social media, the number of online instructional videos has greatly increased. These videos demonstrate how to perform specific tasks such as tying a tie. Understanding the visual information in these types of videos, and the language associated with them is a research problem in the field of machine learning [1]. One method of doing this is to identify key actions in an instructional video, and find the relations between them, as well as to find which objects in the video each step refers to. For the purposes of our project, we refer to a structure containing such information as an action graph.

There are existing techniques for finding such action graphs on videos which are tailored to produce graphs for use in tasks such as action recognition [2,3]. Another kind of technique [1], produces a different kind of action graph. The model uses both the instructional video and associated annotations (transcript) as input. It performs visual grounding (mapping phrases to visual components) with the help of reference resolution (mapping pronouns and implicit actions to phrases). The output is an action graph that represents both the visual grounding and reference resolution information. An example of this kind of action graph is given in Figure 1, although more detailed examples are available in [1].

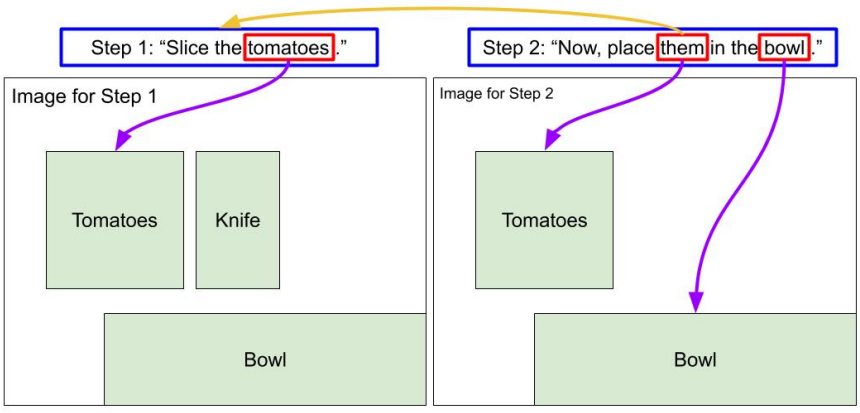

Figure 1: An example of the type of visually grounded action graph used in [1]. Key actions are put in blue boxes. Objects parsed from these actions are in red boxes. The graph consists of connections between objects and actions (reference resolution edges), in yellow, and connections between objects and bounding boxes in the image (visual grounding edges) in purple.

Motivation

Action graphs are a useful representation for instructional videos since they clearly display the dependencies between different steps of the task. For instance, this information can be used to computationally identify which subtasks can be done in parallel.

The issue with the existing class of action graph models meant for action recognition is that many of them focus on low-level actions [2,3]. For instance, a video clip of someone drinking some water would have an action graph consisting of nodes such as “person standing”, “hold cup” and “move cup in front of face” [2]. This is useful in tasks such as action recognition but doesn’t capture relations among higher level instructions.

The action graph described in [1] better captures relations between key instructional steps. We consider it to be the state-of-the-art solution for the problem we are interested in solving. However, it represents connections between steps based on visual grounding and reference resolution, rather than trying to find more general relations. In addition, there is room to improve aspects of the model such as:

- Dataset: The state-of-the-art model is trained using the RoboWatch [6] and YouCookII [7] datasets. The quantity and quality of data present in these datasets may have limited the performance.
- Model Architecture: One realization from the state-of-the-art model was that generalizability was restricted by the visual encoder used and the object proposals method (sub-components of the model) [1].
- Parser: As mentioned in the supplementary material of [1], there are potential inaccuracies introduced by the use of the Stanford CoreNLP parser [5] which was primarily tuned for newspaper datasets rather than instructional text.

Problem Statement

Videos are not directly usable by computational algorithms for tasks such as visual question answering, and learning from demonstration [8, 9]. This is because they contain much extraneous information, and the dependencies between steps are implicit. Action graphs are an alternative representation which solve this problem by explicitly containing the connections between steps and objects. The issue is that many existing algorithms which compute an action graph from a video focus on connections over low-level actions, such as individual movements, rather than higher-level actions, such as the instructional steps themselves.

Given an instructional video, the problem is to produce a high-level action graph, containing key instructional steps and the relations between them. The project will investigate ways of modifying and improving results provided by the state-of-the-art on this task by examining model and dataset variations.

Project Goal

The goal of the project is to produce an algorithm, which takes an instructional video as input, with a time-aligned transcript, and outputs a single visually grounded action graph over the video. More precisely, the nodes of this action graph will contain phrases describing the different steps of the instructional procedure. These phrases will consist of different objects (i.e. items used in the instructional step), and these objects will be connected via an edge to other phrases, as well as to bounding boxes around identified objects in a video frame corresponding to that step. The algorithm is to be compatible with testing on publicly available datasets and is to produce repeatable results on them.

Scope of Work

The project will focus on implementation and design of the algorithm. It is possible that the algorithm will use existing open-source code for required subtasks: such as a pre-trained object proposal network [4], or language parser [5]. The specific components that will be integrated this way are subject to change and will be detailed as a specific implementation of the algorithm develops.

The algorithm will require datasets to learn from, test performance on, and run inference on. Producing such datasets and their annotations is outside the scope of the project, and instead pre-existing ones will be used. However, since the chosen dataset's content and annotation properties have a critical role on the performance of the algorithm, investigating which existing datasets to use will be within the scope of the project. These datasets are also used to evaluate the performance of the algorithm. The design of the evaluation metric for the task is also outside the scope of the project, and pre-existing metrics will be chosen.

A context diagram illustrating the scope of the project is provided in Appendix C.

Requirements Specification (*preliminary*)

Functional requirements, objectives and constraints, as understood at this stage of the project, are listed in Table 1 below.

Table 1: Preliminary project requirements list.

ID	Project Requirement	Description
1	Create a computational algorithm which takes as input an instructional video, and outputs a visually grounded action graph that contains connections between related objects and action phrases in the instruction steps.	**Functional requirement:** This defines the main task of the project.
2	The algorithm's output action graph must be cast into a JSON file.	**Functional requirement:** For inspecting the model output, and for demonstration, the output graph should be cast into JSON format.
3	The algorithm must be able to accept input videos of resolution greater than 144p.	**Functional requirement:** For usability purposes, the algorithm must be able to process input videos of different resolutions, but for decent performance, the resolution cannot be too low. 144p is the lowest resolution available on YouTube.
4	Dataset used must be publicly accessible.	**Constraint:** Dataset accessibility enables others to reproduce the algorithm and measure performance.
5	Must use less than 64 GB of memory space in training.	**Constraint:** Compute clusters such as Google Cloud offer a set of 4 GPUs on a machine, each of which typically have 16 GB of memory [10]. The model must not exceed this memory usage in order to be practically trainable.
6	Minimize space complexity of the	**Objective:** Smaller models are preferable for being

	model.	usable on a wider range of computers.
7	Maximize evaluation metric score across different instructional video domains.	**Objective:** The success of the model is indicated by its performance on the evaluation metric and should be maximized. More successful models will generalize better and obtain higher evaluation scores on different video domains (e.g. on instructional videos about cooking, but also about home improvement).
8	Minimize training time.	**Objective:** Solutions which train in a smaller time are easier to modify and fine-tune for different applications.

Conclusions

Generating action graphs from instructional videos provides a way to distill the information in order to build a computationally useful representation of the video. Our project proposes to generate action graphs by building on existing techniques and aiming to address their limitations. This includes investigating different open source datasets for training and testing. The scope of the work will focus on the design and implementation of the algorithm, whereas existing evaluation metrics and datasets will be used.

References

[1] D.-A. Huang*, S. Buch*, L. Dery, A. Garg, L. Fei-Fei, and J. C. Niebles, "Finding "it": Weakly-supervised, reference-aware visual grounding in in-structional videos", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles, "Action genome: Actions as compositions of spatio-temporal scene graphs", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 236–10 247.

[3] Y.-H. H. Tsai, S. Divvala, L.-P. Morency, R. Salakhutdinov, and A. Farhadi, "Video relationship reasoning using gated spatio-temporal energy graph", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.

[5] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. Mc-Closky, "The Stanford CoreNLP natural language processing toolkit", in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 55–60. DOI: 10.3115/v1/P14-5010. [Online]. Available: https://www.aclweb.org/anthology/P14-5010.

[6] O. Sener, A. R. Zamir, S. Savarese, and A. Saxena, "Unsupervised seman-tic parsing of video collections", in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15, USA:IEEE Computer Society, 2015, pp. 4480–4488, ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.509. [Online]. Available: https://doi.org/10.1109/ICCV.2015.509.

[7] L. Zhou, C. Xu, and J. J. Corso, “Procnets: Learning to segment procedures in untrimmed and unconstrained videos,” *CoRR*, vol. abs/1703.09788, 2017. arXiv: 1703.09788. [Online]. Available: http://arxiv.org/abs/1703.09788.

[8] M. Bhalerao, S. Gujar, A. Bhave and A. V. Nimkar, "Visual Question Answering Using Video Clips," *2019 IEEE Bombay Section Signature Conference (IBSSC)*, Mumbai, India, 2019, pp. 1-6, doi: 10.1109/IBSSC47189.2019.8973090.

[9] *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 2018, pp. 3795-3802, doi: 10.1109/ICRA.2018.8460689.

[10] “GPUs pricing” Google Cloud. [Online]. Available: https://cloud.google.com/compute/gpus-pricing. [Accessed: 22-Sep-2020].

[11] Y. Tang, D. Ding, Y. Rao, Y. Zheng, D. Zhang, L. Zhao, J. Lu, and J.Zhou, “Coin: A large-scale dataset for comprehensive instructional video analysis”, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Appendices

Appendix A: Project Milestones

A list of envisioned milestones until the end of the first semester is provided below in Table 2.

Table 2: Tentative first semester milestones.

Date	**Milestone**
October 26, 2020	As per the supervisor's suggestion, build a trainable model for reference resolution (which will identify connections between text steps in the instructional video). Integrate an evaluation metric for reference resolution. This model should not involve any visual grounding.
November 9, 2020	Use the reference resolution model to build a trainable visual grounding model (which will connect text descriptions of steps to corresponding video frames). Integrate an evaluation metric for visual grounding. Tune the reference resolution model using the visual grounding model.
November 23, 2020	Continue training the previous models. Integrate the models into a larger inference pipeline which can produce an action graph for video inputs. This overall model will be used for the Interim Demo, and will serve as our baseline to make improvements on for later.

Appendix B: Feasibility Assessment

The skills and resources needed, as well as the risks and unknowns, are listed below.

Skills:

- PyTorch: machine learning library in Python used to implement and train models.
- Development over compute clusters: the model training and dataset processing will need to be done remotely on a compute cluster. Accessing and working on these can differ from working on a regular computer.
- JavaScript Object Notation (JSON): open standard file format that will be used to save the output of the algorithm.
- ffmpeg: video processing tool used to change framerate.
- youtube-dl: tool used to download YouTube videos.

Resources:

- We have researched the existing state-of-the-art techniques used in similar tasks and will continue to do so. Research papers and conference videos are accessible on the web.
- We will use publicly available datasets for training the reference resolution and visual grounding models. Potential datasets include the COIN [sic] dataset [11] and the Finding It dataset [1].
- We need computing resources to efficiently train the model. We can use compute clusters available on Google Cloud and those provided by the supervisor.
- Team will provide a commitment of 30 hours/week (10 hours/week/person).

Risks and Unknowns:

- The algorithm has the risk of overfitting to the training dataset which will limit its ability to generalize to other datasets. This should be avoided by using techniques such as cross-validation.
- The model may require a significant amount of dedicated GPU memory which could slow down the training. This must be taken into account while designing and modifying the model architecture.
- Varying model components from the current state-of-the-art [1] may not necessarily improve the performance of the model. However, inferior implementations of the state-of-the-art would still provide results meeting the stated project goal.

Appendix C: System Context Diagram

Figure 2 below contains the system context diagram for the project.

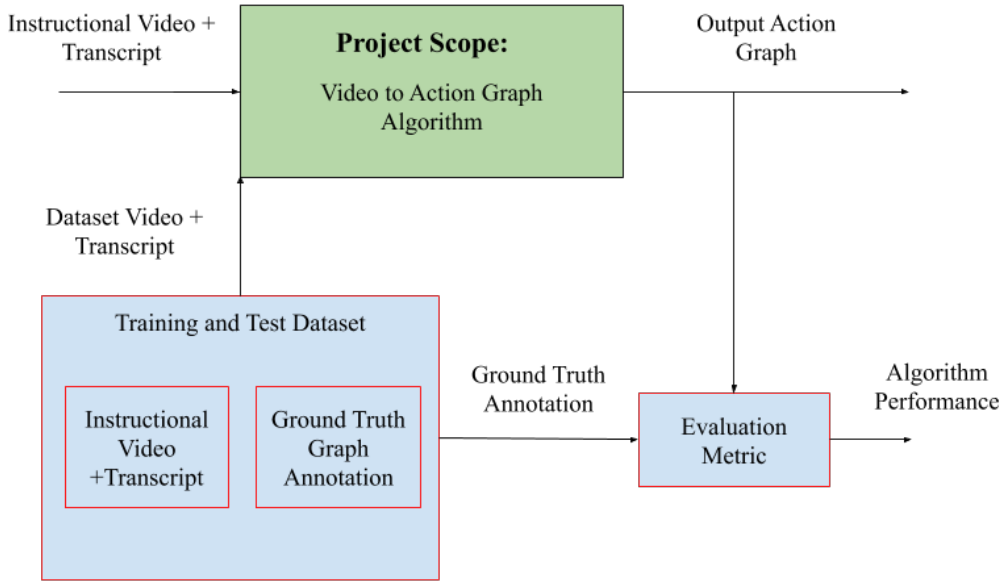

Figure 2: Context diagram of the project. The blue blocks denote components that will be used from other sources. The green block denotes the work the team will produce. Blocks outlined in red are not needed in regular usage of the algorithm, only during training of the model.