NS-Traj: Neuro-Symbolic Human Trajectory Grounding

Mark Endo Joy Hsu Jiaman Li Jiajun Wu

Background

- Commonly, motion capture data is used to learn simple tasks that are limited in complexity, such as action classification or motion forecasting
- Our goal is to develop a method that has a **fine-grain, complex understanding of motion**
- To achieve this goal, we train a model to answer questions that require advanced reasoning over various aspects of human motion

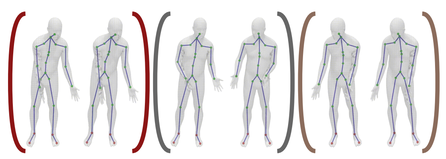
Task Summary

Human Trajectory Question Answering: given a sequence of human motion capture data and a question about the sequence, the model predicts the answer.

Example:

Question: Does the person place down the object on the same side that they pick it up from?

Answer: No

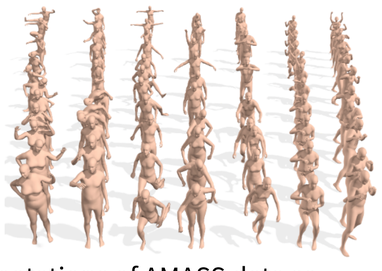


- With these types of questions, we can ground concepts in motion such as actions, adverbs, and body parts
- Using a neuro-symbolic architecture, we hope to generalize to sequences with many actions and complex questions, seeing adverbs in new contexts, and new motion compositions

Data

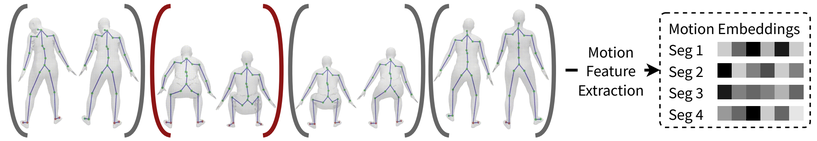
- AMASS[1] is a large database of human motion capture data containing many different types of motions
- HumanML3D[2] provides free-text descriptions of 14,616 sequences from the AMASS dataset
- BABEL[3] contains frame-level annotations of AMASS data so that the sequences can be split up into individual action segments

AMASS



Method

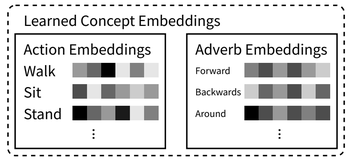
Given a question and corresponding program that were generated from a free-text description of the motion sequence, the model first extracts motion embeddings by passing the segments through a motion encoder. Then, a neuro-symbolic reasoning module executes the program.

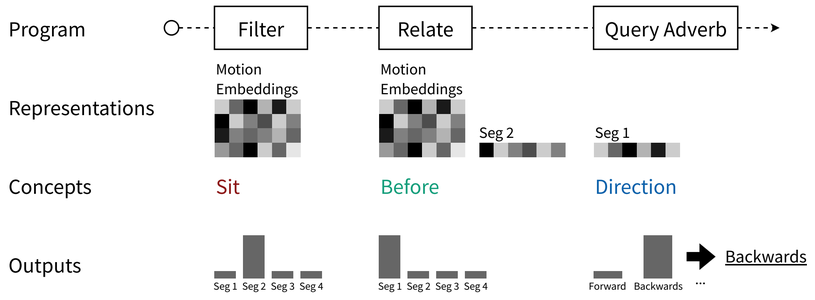


Text: A person walks backwards and sits down then gets back and walks forward.

Question: In what direction does the person move before sitting down?

Program: query_adverb(relate(filter(sit), before), direction)





Question Answer Generation

- Free-text descriptions from HumanML3D can be converted to QA pairs through rule-based templating
- Adverb question example:
  Question: What direction does the person VERB?
  Program: query_adverb(filter(VERB), direction)
  Relevant texts: includes VERB ADV where ADV in [forward, backwards, around, up, down, left, right]

References

[1] Mahmood, Naureen, et al. "AMASS: Archive of motion capture as surface shapes." Proceedings of the IEEE/CVF international conference on computer vision. 2019.
[2] Guo, Chuan, et al. "Generating Diverse and Natural 3D Human Motions From Text." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
[3] Punnakkal, Abhinanda R., et al. "BABEL: Bodies, action and behavior with english labels." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

Results

Training Data	Pre-trained motion encoder	Filter question accuracy	Before query question accuracy	After query question accuracy
Only before/after query questions	No	0.585	0.297	0.262
	Yes	0.656	0.356	0.310
Both filter and before/after query questions	No	0.651	0.336	0.261
	Yes	0.719	**0.365**	**0.333**
Curriculum learning*	No	0.655	0.335	0.273
	Yes	**0.730**	0.360	0.326

* Only train on filter questions for the first half of training, then train on all question types

Example questions/programs:

Before query:

Question: What action is the person doing before walking?

Program: query_action(relate(filter(walk)), before)

After query:

Question: What action is the person doing after walking?

Program: query_action(relate(filter(walk)), after)

Filter:

Question: Which segment is the person walking?

Program: filter(walk)

- Training on all question types and using a pre-trained encoder for motion feature extraction improves performance
- Curriculum learning currently does not offer much benefit, but it may when we increase question complexity

Next Steps

- Complete QA generation from the HumanML3D dataset
- Get results on questions involving adverbs/body parts
- Remove segmenter constraint
- Generate a more accurate dataset of QA pairs using Mechanical Turk

Acknowledgments

I would like to thank Joy Hsu, Jiaman Li, and Professor Jiajun Wu for their invaluable guidance and mentorship throughout the summer. I would also like to thank the CURIS 2022 program for their funding and support.