Action Verb Corpus

Stephanie Gross[1], Matthias Hirschmanner[2], Brigitte Krenn[1], Friedrich Neubarth[1], Michael Zillich[2]
Austrian Research Institute for Artificial Intelligence[1], Technical University Vienna (ACIN)[2]
Freyung 6, Gußhausstrasse 27-29
1010 Vienna, 1040 Vienna
Austria
{stephanie.gross, brigitte.krenn, friedrich.neubarth}@ofai.at, {matthias.hirschmanner,michael.zillich}@tuwien.ac.at

Abstract

The Action Verb Corpus comprises multimodal data of 12 humans conducting in total 390 simple actions (TAKE, PUT, and PUSH). Recorded are audio, video and motion data while participants perform an action and describe what they do. The dataset is annotated with the following information: orthographic transcriptions of utterances, part-of-speech tags, lemmata, information which object is currently moved, information whether a hand touches an object, information whether an object touches the ground/table. Transcription, and information whether an object is in contact with a hand and which object moves where to were manually annotated, the rest was automatically annotated and manually corrected. In addition to the dataset, we present an algorithm for the challenging task of segmenting the stream of words into utterances, segmenting the visual input into a series of actions, and then aligning visual action information and speech. This kind of modality rich data is particularly important for crossmodal and cross-situational word-object and word-action learning in human-robot interactions, and is comparable to parent-toddler communication in early stages of child language acquisition.

Keywords: multimodality, corpus, language modelling

1. Motivation and Related Work

In the following, the multimodal Action Verb Corpus (AVC) is presented, comprising visual and linguistic (German) information related to instances of TAKE, PUT, and PUSH actions. The corpus consists of 140 instances of TAKE/PUT actions each, and 110 instances of PUSH actions). It is geared to robotic action word learning inspired by early word learning of infants. The corpus focuses on modality rich input for crossmodal, cross-situational learning of word-object and word-action mappings. The speciality of AVC is that people perform an action and verbalize what they are doing, leading to rich multisensory data. This is specific to task-oriented communication as well as to parent-toddler communication where the parents' speech is often connected to what the infants see or do (Suanda et al., 2016).

The corpus is small compared to, for instance, the Kinetics video dataset (Kay et al., 2017) comprising more than 300,000 videos of complex actions, or CLEVR (Johnson et al., 2016) comprising a total of 100,000 images of object configurations (locations, object shape, size, color, texture) and approximately a million questions related to those images. While the latter two are designed for training and testing of deep learning models for human action classification such as 'brushing hair', 'riding a bike', etc. (Kinetics), or for diagnosing visual question-answering systems (CLEVR), the Action Verb Corpus focuses on the multimodal representation of instances of TAKE, PUT, and PUSH actions, recorded from adults in order to reflect what a robot will be exposed to when being trained by humans.

In this respect, AVC is complementary to existing child-specific datasets (Nilsson Björkenstam and Wirén, 2013; MacWhinney, 2000). The former collected a longitudinal corpus of parent-child interactions including transcriptions of child-directed speech, child vocalizations, and annotations of gestures, eye gaze, and object-related actions by both parent and child. The CHILDES corpus (MacWhinney, 2000) is a large repository of first language acquisition data in different languages (amongst others German and English), mainly transcriptions of parent and child utterances. Some of the data contain video and audio recordings. Extra-linguistic cues in language learning and the role of social interaction can be investigated based on these data. However, the interactions do not contain the same tasks conducted by different child-adult dyads.

(Gaspers et al., 2014) present a corpus where caregivers/instructors verbally describe learning tasks. Their instructions are - similar to ours - based on visual input. The corpus has information on the wrist position included, so that a robot is able to follow the hand with its gaze. The AVC, in addition, provides information on the arm trajectories and finger joints, and information which object is held by the speaker and where it is moved to.

Three examples for human-robot interaction are the Home Tour Corpus (Green et al., 2006), the Vernissage corpus (Jayagopi et al., 2013), and the Rolland corpus (Anastasiou, 2012). In the first one, users show an environment (a single room, or a whole floor) in a WoZ-Setting to a robot. Recorded were speech, gestures, and gaze. In the second one, a robot induces interactive behavior with and between humans by explaining paintings in a room and then quizzing the participants. In the third one, a user was asked to carry out a set of simple tasks with a powered wheelchair. In the WoZ Setup, the user interacted with the wheelchair via language and gestures. For an overview on annotation tools and schemes, see (Abuczki and Ghazaleh, 2013) or (Tenbrink et al., 2013).

In general, current multimodal corpora typically include language, objects in the visual field, eye gaze, and pointing gestures. An exception is the JCT corpus (Foster et al., 2008) which also includes the currently moved object in object manipulation tasks. Other than in the AVC, the actions were conducted on a screen and not in a real-world setting. These corpora are partially suitable to evaluate multimodal computational models for object-word learning. However, they are not sufficient for crossmodal action learning.

2. Setup for Collecting Data

2.1. Task Scenarios

The overall goal of the data collection activity was to gather multimodal data of basic actions such as TAKE (Ge: *nehmen*), PUT (Ge: *stellen/legen*) and PUSH (Ge: *schieben*). The participants were instructed to perform actions with three objects positioned on a table – a bottle, a can and a box – and verbalize what they are doing. Task 1 was to take an object from the table and put it back on the table at a specific position and use the action verbs *nehmen, legen/stellen* (En: take, put) when verbalizing one's action. Task 2 is comparable to Task 1 but with a focus on verbally specifying the spatial location where the object is put. In Task 3, the user had to push objects to certain locations. The actual actions the user had to perform were displayed in the Virtual Reality headset (Figure 3, right). Sample configurations for Tasks 1 and 2 are shown in Figure 1. In total, there were 4 configurations each depicting a start situation, an intermediate and an end situation. Possible verbalizations for the tasks depicted in Figure 1 would be: T1 – *ich nehme die Flasche und stelle sie neben die Schachtel* ('I take the bottle and put it next to the box'), T2 – *ich nehme die Flasche stelle sie rechts neben die Dose* ('I take the bottle and put it on the right side of the can'), whereby the picture on the left shows the start configuration, the one at the right (2) shows the end configuration, and the one in the middle (1) shows the intermediate configuration where the moved object is not on the table. As regards Task 3, the users were presented with a starting configuration and 10 intermediate configurations indicating which object had to be moved to which position, see Figure 2 for 3 sample moves. Possible verbalizations would be: *ich schiebe die Schachtel rechts neben die Flasche* ('I push the box to the right side of the bottle') (3), *ich schiebe die Flasche zwischen die Schachtel und die Dose* ('I push the bottle between the box and the can') (4), *ich schiebe die Flasche links neben die Dose* ('I push the bottle to the left of the can') (5). While in the TAKE/PUT actions of T1 and T2 the moved object leaves the table (intermediate configuration Figure 1), the pushed object in T3 stays on the table while changing position (configurations 3, 4 and 5 in Figure 2).

2.2. Technical Setup

The user sits (or stands) in front of the table and is wearing an Oculus Rift DK2 Virtual-Reality (VR) headset[1] mounted with the Leap Motion sensor[2] for hand tracking, see Figure 3, left. A camera (Microsoft Kinect) is positioned opposite of the user directed at the table for object tracking. The user performs different actions defined by visual instructions and verbally describes what he/she is doing. Two microphones record the user's description of the performed action. The data is recorded in such a way that it resembles what a robot sees and hears. Letting the human see through the robot's eyes forces the subjects to perform more pronounced movements within a restricted field of vision. This facilitates processing of the input by the artificial system.
The Leap Motion is a stereo infrared camera which is specialized on hand tracking. The Software Development Kit (SDK) provides detailed information on the position of the various joints of the user's arm down to the finger segments. We use the Leap Motion mounted on a VR headset to have the best available tracking performance.

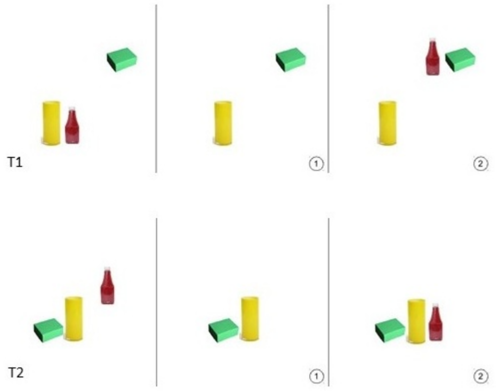

Figure 1: Sample instructions for tasks T1 and T2.

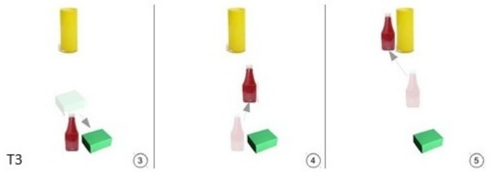

Figure 2: Sample instructions for task T3.

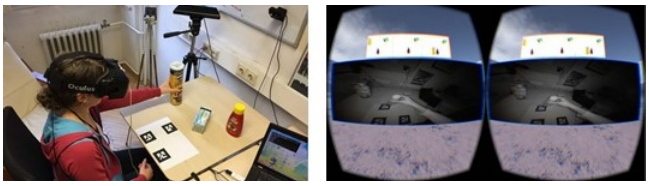

Figure 3: Experimental setup (left) and view through Oculus including instructions (right).

The Oculus Rift DK2 is worn by the user and provides the user's head pose. It is needed to transfer the tracking data of the Leap Motion to a fixed coordinate system. The instructions for the current task are displayed in the Oculus Rift above the camera images of the Leap Motion. This way, the users are able to look at the instructions without moving their heads as it would be the case if they had to look at printout versions of the instructions. Additionally, the setup forces the user to direct the Leap Motion at his/her hands to see what he/she is doing. This behavior is necessary for satisfying hand tracking performance.
For object tracking, the RGB as well as depth data of the

[1] https://www.oculus.com/dk2/

[2] https://www.leapmotion.com

Kinect camera is recorded as a ROS[3] bag on a separate machine running Ubuntu. The models of the objects were created beforehand with the RTM-Toolbox from the V4R library[4]. The objects were positioned on a turntable and a Kinect camera recorded a sequence of RGB-D images. The software tool uses a Kanade-Lucas-Tomasi feature tracker with FAST features to create an object model as described in (Prankl et al., 2015). The features for creating the model are saved with the model and used for object tracking.

We apply the object tracker from the V4R Library on the recorded data from the user experiments[5]. It utilizes the Perspective-n-Point algorithm to calculate the 6-Degrees-of-Freedom pose of the objects from a monocular image stream. The offline tracking enables the best possible results because the object tracker can be tuned for a specific recording. Besides the position and orientation of the object, two Boolean variables are saved: object is in contact with the table and object is in contact with a hand. The former is set automatically depending on the object's position, the latter is currently manually annotated. We use the following information to process the raw data from the object tracker: If the object is in contact with the user's hand, the object position is saved as received from the object tracker. If the object is not in contact with the hand, we calculate an average object pose because we assume that the object does not move. This assumption is correct in our experimental setup, as long as the user does not push an object with a different object. The weighted arithmetic mean is used whereby the weight is the confidence received from the object tracker. Tracked object positions with a low confidence or further away than 10 cm from the average position are discarded to omit wrong positives. However, we count the number of outliers with a high confidence. If this number exceeds a certain threshold, the current average position is discarded. This way, the jittering of the raw object position is cancelled out and occlusions do not impair tracking performance if the object was successfully tracked before.

The recorded data from the different systems (hand and object tracking) are transformed to a common coordinate frame (fixed in the user's head) and temporally aligned in a post-processing step. This alignment is done manually.

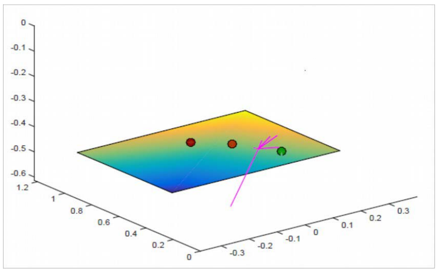

Figure 4: Visualization of manipulating hand and objects.

3. Dataset and Annotations

The corpus, at the time of writing, comprises 140 instances of TAKE/PUT actions each and 110 instances of PUSH actions from 20 recordings of Task 1, 15 recordings of Task 2 and 11 recordings of Task 3. See Table 1 for an overview. Overall, 12 persons conducted the tasks.

Task	Number of Recordings	Number of Actions per Recording	
T1	20	4	TAKE/PUT-actions
T2	15	4	TAKE/PUT-actions
T3	11	10	PUSH-actions

Table 1: Composition of corpus.

3.1. Representation of Information

Apart from the raw data from hand and object tracking, the audio recordings, and grey-scale videos from the user perspective, each instance of a task is represented by:

1. the merged output of the hand and the object tracker, including per frame the 3D positions of the joints in the elbow, wrist, and knuckles of the instructor's hands as well as the object poses and their reliability estimate calculated by the object tracker;
2. an animation of the merged hand and object tracking, see Figure 4;
3. manual orthographic transcriptions of speech: one is close to what is actually spoken, the other one is normalized with respect to standard NLP tools such as taggers, parsers, stemmers, etc.;
4. part-of-speech tags, automatically generated with the Tree-Tagger (Schmid, 1995) and manually corrected;
5. lemmata, automatically generated with the Tree-Tagger and manually corrected;
6. the information which object is moved and where it is moved to (manually annotated);
7. the information whether the left or right hand touches a particular object (manually annotated);
8. the information whether a particular object touches the ground/table (automatically identified by the object tracker and manually corrected);
9. segmentation of the stream of words into chunks using heuristics such as long pauses (min. 0.5 sec) or connectives (e.g. *und* – 'and');
10. segmentation of the visual input into situations comprising actions: in our specific dataset, a situation is construed each time a hand touches an object and the object moves. If the object is lifted from the ground, the situation consists of the two actions TAKE and PUT, otherwise it is a PUSH action.

While the information described in 1. is provided as a CSV file per recording, the information described in 3. to 10. (henceforth the annotation tiers) is represented as Elan[6] file (.eaf) and CSV exported from Elan. All annotation tiers are synchronized with the real-time animation of the hand and object tracking and with the speech stream. Praat[7] is used

[3] http://www.ros.org

[4] http://www.acin.tuwien.ac.at/forschung/v4r/software-tools/rtm/

[5] http://www.acin.tuwien.ac.at/forschung/v4r/software-tools/v4r-library/

[6] https://tla.mpi.nl/tools/tla-tools/elan/

[7] http://www.fon.hum.uva.nl/praat/

for transcribing the utterances.[8]

3.2. Alignment of Visual Action and Speech

For tasks such as crossmodal learning of word-object and word-action references, the chunks obtained from the transcribed speech as described in 9. above, and the actions retrieved from the visual representation of situations (see 10. above) need to be aligned. Each situation – a multimodal perceptual frame – comprises one or two events. An event consists of an action (with all involved objects) and an utterance that verbally describes what happens. In order to obtain these events, information from different modalities has to be processed in order to identify actions, utterances and correct alignment between them.

For the given dataset, each action ideally is described by exactly one utterance (made up of one or several chunks), see for instance the following example where the speech sequence *ich schiebe die Dose* ('I push the can') is followed by a (relatively long) pause of 2.082 seconds and then followed by the sequence *neben die Schachtel* ('next to the box'). These two chunks need to be aggregated into one utterance and aligned to the corresponding action. Alignment proceeds in two steps: first the right boundary is identified from the set of chunks, in a second step the algorithm works backwards in order to determine the left boundary. When a situation consists of two actions, the string of selected chunks has to be split into two utterances, otherwise the string of chunks directly maps to a single utterance.

In order to identify the rightmost speech chunk that is to be aligned to the current action(s), the algorithm proceeds left to right on the list of chunks. The endpoint of the situation (plus some latency, here set to 0.7 sec) marks the preliminary cutoff point (1), see Figure 5. Chunks that start later than that can be connected to the preceding chunk, if the pause between the previous and the currently examined chunk does not exceed a given threshold of 1.4 sec (2). In addition to that, the chunk must not overlap with the next situation (3) and must not start with a connective ('and') since this would hint to the begin of a new utterance (4).

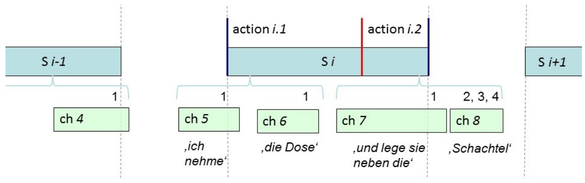

Figure 5: Step 1: Alignment of utterance and action.

In the second step, starting from the last chunk, preceding chunks are merged until either the chunk ends long before the situation starts, or it is already aligned to a preceding situation. The threshold for the difference between endpoint of speech chunk and begin of situation is set to 7 sec, equivalent to temporal bounds in 'working memory'(Miller, 1956). Only if there are two actions within a given situation ('take' together with 'put') – implying that there should be two utterances describing theses actions – for the second utterance the threshold is set to 1.4 sec. In that case, the algorithm is also attentive to continuation words ('and'). When it hits one, it would regard this chunk as the begin of the second utterance. See Figure 6.

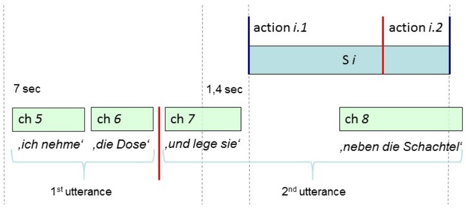

Figure 6: Step 2: Alignment of utterance and action.

This procedure presupposes that speakers tend to start the description of an action before executing it. This presupposition is supported by the empirical facts. The accuracy of the algorithm is high (92% on a subset of the AVC and 86% on Dataset 1 of the MMTD corpus[9]), and it also guarantees that explanations and comments by the test persons that are temporally outside the given bounds and do not pertain to the actions described, are not aligned to actions.

4. Conclusion and Future Work

So far, the presented corpus provides a solid basis for crossmodal and cross-situational word-object and word-action learning, inspired by early language acquisition in children. In addition, we presented an algorithm for segmenting actions and utterances, and aligning the two. The three actions TAKE, PUT, and PUSH can be successfully segmented by the presented heuristics, however, there are limitations as the algorithm is fine-tuned to the given dataset. As we plan to extend the number and complexity of actions, we expect the list of relevant features to segment those actions to be extended. In addition, some aspects of visual actions are currently manually annotated (whether an object is moving and whether a hand touches an object). Work is underway in order to recognize this information automatically. An open issue is also the granularity of actions (e.g., whether the GRASP action preceding a PUSH action should be modeled as part of the PUSH action, or as a separate action) and the effect this has on crossmodal and cross-situational action learning in robots.

5. Acknowledgements

This research is supported by the Vienna Science and Technology Fund (WWTF), project RALLI – Robotic Action-Language Learning through Interaction (ICT15-045) and the CHIST-ERA project ATLANTIS (2287-N35). The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry for Science, Research and Economy.

[8] The dataset (comprising the information described in 1. to 10.) can be downloaded from `http://www.ofai.at/research/interact/avc.html`.

[9] `http://www.ofai.at/research/interact/MMTD.html`

6. Bibliographical References

Abuczki, Á. and Ghazaleh, E. B. (2013). An overview of multimodal corpora, annotation tools and schemes. *Argumentum*, 9:86–98.

Anastasiou, D. (2012). A speech and gesture spatial corpus in assisted living. In *LREC*, pages 2351–2354.

Foster, M. E., Bard, E. G., Guhe, M., Hill, R. L., Oberlander, J., and Knoll, A. (2008). The roles of haptic-ostensive referring expressions in cooperative, task-based human-robot dialogue. In *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*, pages 295–302. ACM.

Gaspers, J., Panzner, M., Lemme, A., Cimiano, P., Rohlfing, K., and Wrede, S. (2014). A multimodal corpus for the evaluation of computational models for (grounded) language acquisition. In *Proceedings of EACL Workshop on Cognitive Aspects of Computational Language Learning*, pages 30–37.

Green, A., Hüttenrauch, H., Topp, E. A., and Eklundh, K. S. (2006). Developing a contextualized multimodal corpus for human-robot interaction. In *Proceedings of the 5th International Conference on Language Resources and Evaluation LREC*. Citeseer.

Jayagopi, D. B., Sheiki, S., Klotz, D., Wienke, J., Odobez, J.-M., Wrede, S., Khalidov, V., Nyugen, L., Wrede, B., and Gatica-Perez, D. (2013). The vernissage corpus: A conversational human-robot-interaction dataset. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 149–150. IEEE Press.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. B. (2016). CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890.

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

MacWhinney, B. (2000). The childes project: Tools for analyzing talk: Volume i: Transcription format and programs, volume ii: The database. *Computational Linguistics*, 26(4):657–657.

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81.

Nilsson Björkenstam, K. and Wirén, M. (2013). Multimodal annotation of parent-child interaction in a free-play setting. In *Thirteenth International Conference on Intelligent Virtual Agents (IVA 2013)*.

Prankl, J., Aldoma, A., Svejda, A., and Vincze, M. (2015). Rgb-d object modelling for object recognition and tracking. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 96–103. IEEE.

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*. Citeseer.

Suanda, S. H., Smith, L. B., and Yu, C. (2016). The multisensory nature of verbal discourse in parent–toddler interactions. *Developmental Neuropsychology*, 41(5-8):324–341.

Tenbrink, T., Eberhard, K., Shi, H., Kuebler, S., and Scheutz, M. (2013). Annotation of negotiation processes in joint-action dialogues. *Dialogue & Discourse*, 4(2):185–214.