Deep Episodic Memory for Verbalization of Robot Experience

Leonard Bärmann, Fabian Peller-Konrad, Stefan Constantin, Tamim Asfour, Alex Waibel

***Abstract*—The ability to verbalize robot experience in natural language is key for a symbiotic human-robot interaction. While first works approached this problem using template-based verbalization on symbolic episode data only, we explore a novel way in which deep learning methods are used for the creation of an episodic memory from experiences as well as the verbalization of such experience in natural language. To this end, we first collected a complex dataset consisting of more than a thousand multimodal robot episode recordings both from simulation as well as real robot executions, together with representative natural language questions and answers about the robot's past experience. Second, we propose and evaluate an episodic memory verbalization model consisting of a speech encoder and decoder based on the Transformer architecture, combined with an LSTM-based episodic memory auto-encoder, and evaluate the model on simulated and real data from robot execution examples. Our experimental results provide a proof-of-concept for episodic-memory-based verbalization of robot experience.**

***Index Terms*—Learning from Experience, AI-Enabled Robotics, Natural Dialog for HRI, Sensorimotor learning**

I. Introduction

HUMANS have the ability to retain information over time by encoding, storing and retrieving information in a complex memory model. In humans, episodic memory is concerned with the recollection, organisation and retrieval of episodes, i. e. personally experienced events or performed activities occurring at a particular time, place and context. Such recollection of events and activities plays a key role for the acquisition of semantic knowledge in a cognitive system. Our episodic memory allows us to reason about past events, to argue why we acted in a particular way, or to recall what problems occurred in the past. The externalization of experienced episodes, i. e. the content of the episodic memory, is key for interaction and communication between humans and robots as this strongly influences usability, transparency and acceptance of the robot by the user [1]. In this work, we present an episodic-memory-based verbalization of a humanoid robot's experience using natural language (Fig. 1). We consider daily household activities where a humanoid robot performs tasks in a kitchen environment such as cleaning the table or loading dishes into the dishwasher. Using the developed verbalization model, the robot should be able to answer questions asked by the user about tasks performed by the robot during the day, e. g. "Did you see the red cup?", or describe important events and failures that happened, e. g. "I tried to put the juice into the fridge, but the door was blocked by the chair".

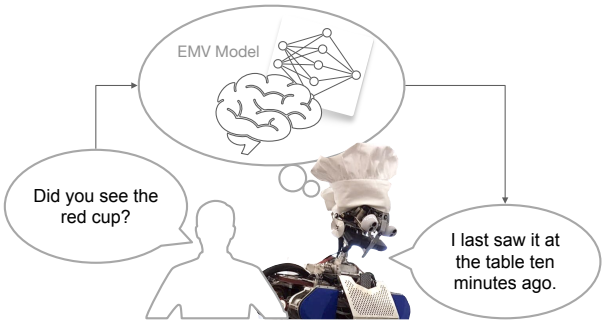

Fig. 1. Episodic memory verbalization

To perform the aforementioned tasks, the robot needs to have a component comparable to the human episodic memory, encoding, storing and recalling experiences [2], [3]. In this context, an experience can be seen as a snapshot of the internal state of the robot consisting of the robot's camera images in ego-centric view, the current configuration of the robot, the estimated robot position as well as all detected objects and their estimated poses, detected human poses, the current executed action with its status and arguments and the current task of the robot as given by the symbolic planner. Every experience is enriched with corresponding timestamps. Thus, an episode can be seen as a temporally ordered collection of such experiences concerned with achieving a single planner goal.

To address the problem of robot experience verbalization, we propose a data-driven method employing deep neural architectures for both creating an episodic memory from experiences as well as the verbalization of such experience in natural language.

Our first main contribution is the collection of a multimodal dataset of robot experiences, consisting of the aforementioned information (both symbolic and sub-symbolic). In an automated way, we collected more than a thousand episode samples from simulation. Further, we performed a data collection on the humanoid robot ARMAR-III [4] in a real kitchen environment in order to evaluate our method on realistic data. For training, recordings are annotated with natural language questions and answers using a grammar-based dialog generator, while this annotation is performed by humans for final system evaluation.

Manuscript received: December, 21, 2020; Revised March, 24, 2021; Accepted May, 14, 2021.

This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments. This work has been supported by the German Federal Ministry of Education and Research (BMBF) under the project OML (01IS18040A).

The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany. leonard.baermann@student.kit.edu, {fabian.peller, stefan.constantin, asfour, waibel}@kit.edu

Digital Object Identifier (DOI): see top of this page.

Second, we propose a novel neural architecture for episodic memory verbalization (EMV). The inherent sequential nature of episode data suggests the use of an LSTM [5] encoder [6] network to process the stream of robot experiences. Latent representations of this stream, referred to as the episodic memory (EM), are then used to provide contextual information to the verbalization task, handled by a Transformer network [7]. Specifically, we employ the pretrained T5 architecture [8] for natural language processing. We open-source our dataset, code and models[1].

The remainder of this paper is structured as follows: We discuss related work in section II. In section III, we introduce our approach and explain the details of the Episodic Memory Verbalization Model. In section IV, we describe the requirements and procedure for data collection. In section V, we present evaluation results of the EMV model. Finally, section VI draws a conclusion and describes future work.

II. Related Work

We first discuss previous work on the implementation of verbalization and episodic memory systems in robotics and then proceed with the relation of this work to the area of video question answering and task-oriented dialog.

A. Robot Experience Verbalization

The first work to introduce the concept of *verbalization* for autonomous robots was [1]. The authors propose a system that allow the "CoBot" robots to narrate navigation routes through a building using natural language. To achieve context adaptability, the authors define the concept of *verbalization space*, constituted of the (discrete) dimensions *abstraction*, *locality* and *specificity*. The variable verbalization algorithm is a rule-based procedure that receives the route and map information as well as a point in verbalization space as input and generates a natural language description of the route according to the given verbalization preferences. In their follow-up work [9], the authors address the automatic determination of the requested point in verbalization space from a given user query. For example, the request "Please tell me exactly how you got here." would map to detailed narrative, semantic abstraction level and global locality. For this purpose, a corpus of 2400 queries was collected using online crowdsourcing. Then, classic machine learning algorithms like Naive Bayes classifier were used, with results showing a classification accuracy of about 70 %, which is stated to be sufficient since the model can also be used to refine the verbalization parameters.

Both previously introduced works are concerned with verbalization, but do not explicitly address the connection to an episodic memory system. To address this problem, [10] construct an experience-based verbalization system for navigation and manipulation tasks using the ARMAR-III kitchen robot [4]. Here, episodic memory is realized as a *Log File Generator*, which records significant events during robot program execution at different levels of abstraction. To narrate past episodes, the rule-based variable verbalization algorithm introduced in [1] is adapted to the given scenario. Eventually, the system is able to give narrations like "I picked up the green cup on the sink with my right hand. It took 4 seconds.".

B. Episodic Memory

Verbalization of robot experiences requires an episodic memory that allows encoding and recalling of experienced events and performed activities. Since the introduction of term in 1972 [2], various works in robotics dealt with transferring the concept of EM to robotics. For instance, the CRAM_m system [11] approaches the problem using a semantic database. On the one hand, plan events (including time information) are asserted to the predefined, information-rich KnowRob ontology [12] when actions are executed. On the other hand, low-level perceptual data is recorded in a NoSQL database, including camera images as well as robot pose information captured at important points during plan execution. Projecting the sub-symbolic data into the ontology enables the combined system to answer various queries about past episodes; however, these questions need to be given as Prolog queries.

Recently, we introduced an EM based on deep neural networks to encode a robot's action experience [3]. This Deep EM learns latent space representations from videos in an unsupervised manner using an encoder-decoder architecture. First, an input sequence of video frames is passed to a convolutional LSTM network, yielding its final hidden and cell state as latent vector V. In the following, two inverse convolutional LSTM decoders receive V, one with the goal of reconstructing the input sequence and another one predicting the next few frames of the video. The network is trained with a combination of image reconstruction and gradient difference loss using two large-scale video datasets containing videos of human manipulation activities. The resulting multi-functional EM, i. e. the collection of latent space vectors, can be used to group similar episodes, perform action recognition (if existing EM entries are labeled), predict the next frame, or learn object manipulation.

Similarly, in [13], a MaskRCNN-based deep learning model called RobotVQA is trained on robot-perspective images including a depth channel, however with the goal of inferring semantic scene information instead of constructing episodic memories. With the focus on learning robot actions from demonstration, [14] use a deep adaptive resonance theory (ART) neural model instead of a sequence-to-sequence model to learn an EM model. The EM in [15] is also based on an ART neural model and enables efficient information retrieval using various types of cues, with the memory consumption controlled by a mechanism of gradual forgetting. In the ART-based neural model of [16], time is explicitly modeled and information can also be retrieved given a time interval.

Furthermore, video embeddings using temporal cycle consistency learning, as constructed in [17], can be seen as episodic memories of a video.

C. Question Answering

EMV is related to Question Answering (QA) as it is a special case thereof. For instance, Video Question Answering

[1] https://gitlab.com/lbaermann/verbalization-of-episodic-memory

(VQA) systems can be seen as constructing an EM from video experiences. A video sequence as well as a natural language question about an episode is provided and the model is supposed to output the corresponding answer. VQA task complexity varies from choosing among a multiple-choice-set of answer sentences [18] to choosing a single output word [19] up to open ended answers sentences constructed using a decoder network [20].

If the episodic memory is made up of symbolic data only, similarities to fact-based QA like in [21] arise. Here, the EM is constructed from a set of natural language facts, which is then used to respond to a question. Similarly, in Task-Oriented Dialog (TOD), recent works like [22] present differentiable architectures to understand natural language questions and access symbolic knowledge bases to give appropriate answers.

While hidden representations of such (V)QA networks might be considered implicit episode representations, the major difference to EMV is question timing. In QA, the video (or supporting facts or knowledge base) and the corresponding query are given to the model simultaneously, so that e.g. attention mechanisms [23] can attend to parts of the video specifically relevant to the question. However, in an EMV system, the user's question about an episode might occur at an arbitrary point later in time (e.g. consider the query "What did you do yesterday?"). Thus, the input to EM cannot be analyzed with respect to the question directly. Instead, explicit episode representations have to be constructed and stored in advance, while the raw input stream of episodic experiences is discarded.

Furthermore, the field of language command grounding for Human-Robot-Interaction, as in [24], [25], is related since it tries to connect natural language content with multi-modal representations of the robot's environment.

This work differs to the existing literature by 1) using deep-learning methods for both understanding natural language questions and generating answers (in contrast to [9], [10]), 2) requiring an explicit, question-independent representation of EM (in contrast to the QA works, as explained above), 3) using a temporal stream of episode data in contrast to a static knowledge-base in TOD works and 4) combining different modalities to build up and verbalize an episodic memory (in contrast to and as an extension of [3], [10]).

III. Approach

We present the Episodic Memory Verbalization (EMV) model architecture, which is constituted of three major components, see Fig. 2. The *EM encoder* processes a multi-modal stream of episodic data and creates a latent representation of episodic memory (EM) from raw experiences. A compact representation of the robot's past experience is achieved as the model significantly compresses the EM content to a low-dimensional latent representation compared to the extremely high-dimensional, multi-modal input. Ideally, the EM would be constrained to a constant size. However, for the sake of simplicity, we allow the EM to grow linearly with the number of key frames fed into the episode encoder, and we will report on the more complex "$O(1)$ EM" task, which additionally requires controlled forgetting of irrelevant information in the EM, in future work.

The second major component is the *speech encoder*, receiving a natural language question about the robot's past as well as the timestamp of this question. We simplify the following explanations by defining a *query* to be a natural language question prepended with a textual representation of the question timestamp, e.g. "2020 12 10 4 14 56 07 What did you do yesterday before noon?".

Resulting activations from the speech encoder are then passed to the third component, the *speech decoder*, which combines the query encoding with the EM content and outputs the natural language answer. In the following, we describe each model component, as also depicted in Fig. 2.

A. EM Encoder

The inherent sequential nature of episode data given as a temporal sequence of key frames suggests the use of an LSTM network [5] for creating the EM. Each key frame corresponds to one input time step for the LSTM. However, key frames contain highly multi-modal information, and therefore first have to be converted to a vector encoding and embedded into a common space. For this purpose, we define four different data parts for one key frame:

1) **Timestamp**. It is encoded as a seven-dimensional vector, with the dimensions corresponding to year, month, day, day-of-week, hour, minute and second, respectively.
2) **Symbolic information**. Using a dictionary built up during pre-processing, we represent this data as a vector, where each dimension corresponds to one of the following entries: task goal provided by the planner, task execution status (success, failure, abort), action, action arguments, action execution status (started, success or failure), object name (ID) and detection status. Unfilled entries are set to a special padding token.
3) **Subsymbolic information**. These numeric values are simply concatenated to form a long vector. Included data are: detected object positions for up to two objects (in the robot's root coordinate system), platform position (world coordinate system), human pose data as provided by OpenPose [26] (camera coordinate system), and kinematic data of the robot's joints (angles, velocities and currents). Every dimension of this subsymbolic vector is normalized with respect to its mean and standard deviation in the training set.
4) **Pre-Encoded Latent vector**. We create a latent representation of the image associated with each key frame as produced by the Deep EM network of [3]. This auto-encoder model is trained as described in the original paper and fine-tuned using our recordings. However, it is not trained together with the EMV model, i. e. the produced representations are ordinary input data to our model.

Each of the four data parts defined above is passed to a fully connected layer separately to either expand (for symbolic data) or reduce (for subsymbolic data) the dimensionality. All resulting embeddings are then concatenated and serve as the

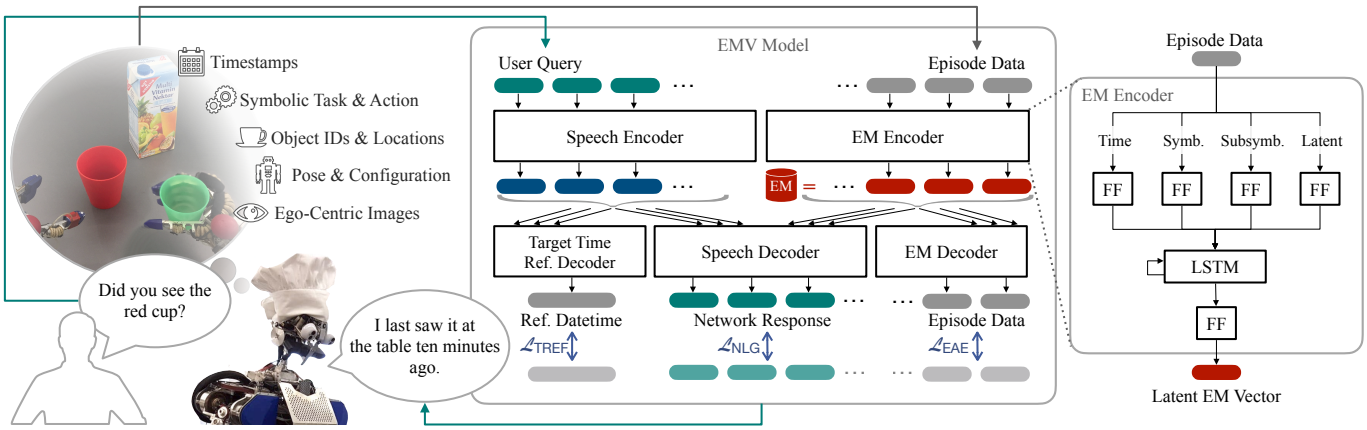

Fig. 2. An overview of the Episodic Memory Verbalization system. The EMV model continuously receives episode data consisting of multi-modal data streams. This sequence of episode key frames is passed through the EM encoder network to build a latent representation to be stored in the episodic memory. The EM is the collection of latent representations produced by the EM encoder. At some point later in time, a user asks a question about the robot's past, which is then given to the speech encoder, including the date and time at which the question is asked. In the following, the decoder attends to both the query encoding as well as the latent EM content to produce the natural language response, which is finally sent to the robot's text-to-speech component. The model is trained end-to-end, with two supplementary decoders to provide auxiliary contributions to the overall loss function.

input for the episode encoder LSTM. The hidden states of this encoder are again passed through a fully connected layer to produce the latent EM vector for each time step (see the right part of Fig. 2).

The EM itself is the list of all latent representations produced by the EM encoder, given a sequence of input key frames. While this behaves like an end-to-end model during training (and batched evaluation), it does not do so when deployed to a robot (where there might be an indefinite amount of time between an experience and a question, i. e. the EM vectors are stored long-term and retrieved when a question occurs). Thus, our contribution is not the direct creation of a large EM from the training data. Instead, our contribution is to create a learned model which can produce latent representations of EM when deployed to a real robot.

B. Speech Encoder and Decoder

For the natural language processing part of the EMV model, we use the Transformer architecture as introduced in [7]. This sequence-to-sequence model is based on repeatedly applying attention to its own hidden states. For mathematical details on the model, please refer to [7]. Specifically, we make use of the "small" T5 model, as proposed by [8], pretrained on both an unsupervised fill-in-the-gap task as well as supervised sequence-to-sequence language tasks (e. g. translation). The speech encoder of the EMV model, which receives the input query, is identical to the original encoder of the T5 network. Its resulting activations, referred to as the query encoding, are then passed to the decoder, which additionally receives the sequence of latent EM vectors. This means, the T5 decoder, which itself is also unchanged as defined in [8], receives a longer sequence of hidden states, through which it can attend to both query and EM content simultaneously. Finally, the decoder yields a distribution over the vocabulary, thus generating the response word by word.

C. Supplementary Decoders & Loss Function

With the components described above, the model could be trained with cross-entropy loss based on the natural language generation (NLG) output, as usual in sequence-to-sequence tasks [8]. However, to facilitate and analyze learning of the EMV task, we add additional decoders and additive loss contributions.

Inspired by the auto-encoder loss for Deep EM in [3], the model is enriched with an episode decoder. It receives the output of the EM encoder, i. e. the sequence of latent EM vectors, to map it back to its original input, i. e. the sequence of key frames. For this purpose, we use a straightforward approach, with four separate linear layers all receiving the latent EM vectors (step-by-step). The first two map to a probability distribution over all possible date-time values or EM tokens (symbolic tokens from dictionary), respectively, separately for each date-time or symbolic episode entry. These two parts are trained as a classification task, i. e. with cross-entropy loss. The third and fourth linear layer directly try to regress the sub-symbolic and latent part of the episode data, using MSE and L1 loss functions, respectively. Experiments were performed for training the complete model with NLG and episode auto-encoder (EAE) loss simultaneously, as well as pretraining the EAE separately before integrating it into the EMV model.

To analyze the model's capability to properly detect and extract date-time references in query strings, we add an additional *time-reference decoder* to the model. It attends to the hidden states from the query encoder (using a primitive version of attention mechanism as introduced in [23]), and is supposed to output the date-time referenced in the query. The target values, i. e. the correct timestamp references, are available because we generated our training dialog dataset automatically, as explained in section IV.

To summarize, the complete loss function $\mathcal{L}$ for training the EMV model is $\mathcal{L} = \alpha_1 \mathcal{L}_{NLG} + \alpha_2 \mathcal{L}_{EAE} + \alpha_3 \mathcal{L}_{TREF}$ with

$\mathcal{L}_{NLG}$, $\mathcal{L}_{EAE}$ and $\mathcal{L}_{TREF}$ being the loss from natural language generation, episode auto-encoder and time-reference reconstruction, respectively. The EAE loss is $\mathcal{L}_{EAE} = \beta_1\mathcal{L}_{DT} + \beta_2\mathcal{L}_{Sym} + \beta_3\mathcal{L}_{Sub}$ where $\mathcal{L}_{DT}$ is the loss of the date-time reconstruction (classification), $\mathcal{L}_{Sym}$ is the classification loss on the symbolic information entries and $\mathcal{L}_{Sub}$ is the regression loss on the subsymbolic key frame content. α_i and β_i are hyperparameters, with $\sum_i \alpha_i = \sum_i \beta_i = 1$.

D. Training Procedure

The first step of training is to learn a simple dimensionality reduction for the latent image vectors as produced by the Deep EM [3]. We use a non-linear neural network with a single hidden bottleneck layer, trained as an auto-encoder, for this purpose. The latent vectors are then pre-processed to the reduced dimension and thereafter treated as a regular part of the input and target data.

Second, the episode dataset is used to pre-train the EM encoder and decoder as an episode auto-encoder (i. e. using only $\mathcal{L}_{EAE}$ from above). During that, the data is regenerated after each few epochs, choosing new random date-times for the episodes. This way, the variety of date-time inputs the model has seen is increased, preventing overfitting and aiding to generalization.

As a third and last step, the complete EMV model is trained, initializing the EM encoder and decoder with the weights from episode auto-encoder pre-training. We apply a curriculum learning strategy during this step, starting with histories of one episode, and increasing to histories of up to five episodes. Because of the large number of parameters of the pre-trained T5 speech model, we train only the speech encoder and the first layer of the decoder. Most importantly, we avoid training T5's output (fully-connected) layer, making up the biggest number of parameters.

IV. Dataset Collection

We collected a dataset for the EMV task consisting of 1) robot episode recordings (RER) and 2) natural language questions and answers corresponding to the RERs.

An RER is defined as a stream of symbolic information (task, goal, action and action arguments as stated by the symbolic planner), action execution status (started, running, interrupted, finished with success or failure) and detected objects IDs as well as subsymbolic data (camera images, robot configuration, estimated platform position, estimated object poses, detected human poses) and timestamp. To prevent overwhelming our model by high-frequency data streams, we extract key frames from an RER. For this, we subsample from the subsymbolic data stream with a time window of five seconds, e. g. we keep only the latest robot pose and position for each five seconds. However, whenever a new symbolic action has been triggered or the action state changes, a new time window starts immediately (i. e. all data available to the memory at that moment will be stored). As already described in section I, we define a single episode to be a contiguous sequence of key frames, concerned with achieving a single planner goal. This implies that an RER corresponds to a sequence of one or more episodes.

For training a deep neural model, gathering large amount of data is crucial. Therefore, we use the simulation component of the ArmarX robot framework [27] to randomly create scenes with different objects at different places in a kitchen environment. Then, an episode recorder component is used to capture single-episode RERs, automatically iterating over all created scenes and possible planner goals. For our exploration study, the goals were of the form $objectAt(o_i, l_n)$, $objectAt(o_i, l_n) \wedge objectAt(o_j, l_m)$ or $grasped(h_k, o_i)$ with hand $h_k \in \{\text{left}, \text{right}\}$, objects $o_i, o_j \in O, |O| = 3, o_i \neq o_j$ and locations $l_n, l_m \in L, |L| = 5$. To further increase the size of our dataset, we employ three data augmentation techniques:

- Each single-episode RER is copied multiple times, adding random noise to subsymbolic data where appropriate (joint angles, human pose data, object and robot position).
- Each episode can be moved to any point in time by simply adjusting all contained timestamps uniformly. This step is crucial for building a representative training dataset containing a wide variety of possible date and time values.
- Multi-episode RERs are built by concatenating random combinations of the simulated single-episode RERs. This also includes moving each to a different timestamp, again. In the rest of this paper, the term *history* will refer to such a multi-episode RER, and the *length* of a history is the number of episodes it is constructed from.

We split the set of simulated RERs into train, valid and test split randomly *before* performing dataset augmentation separately on each split to avoid any systematic error.

In addition to episodic data, natural-language QA pairs have to be acquired. To efficiently make use of our large number of simulated RERs (1011 in total), we implemented a grammar-based QA generation script, creating thousands of possible questions and answers per episode. The script considers the symbolic information in the recordings, randomly chooses a question timestamp (after the last episode), and then conditionally generates possible dialogs, spanning different levels of complexity from "What did you do? – I moved the green cup to the round table" to "How long exactly did it take to release the red cup at 03 PM? – It took 26 seconds". The hand-written grammar is inspired by the results of a precedent small-scale human QA data collection (Using grammars to generate dialog datasets is a common approach, see e. g. [28]).

As mentioned before, the collected dataset also includes a test set to evaluate the performance of our model. We will refer to this test set of simulated RERs with grammar-generated dialogs as *simulated-robot-grammar-generated*. To assess various levels of generalization capability, we use several additional test sets: 1) The *simulated-robot-grammar-generated* test set also includes histories with a larger number of episodes than seen during training. This way, we can check how well the EM generalizes to longer sequences. 2) Using an independently collected, small set of simulated RERs, we asked humans to annotate the data by entering questions and answers related to the EMV task, given the video of the recording and randomly chosen timestamps of the episode and

the question via a newly created web interface[2]. This serves for assessing how well the language model generalizes to dialog constructs unseen during training. We refer to this test set of simulated RERs with human-entered questions and answers as *simulated-robot-human-annotated*. 3) To evaluate generalization ability beyond robot simulation, we additionally collected a small sample of RERs on a real robot, using the same episode recorder component already employed in simulation. QA-annotation was done using the dialog generator as well as by employing human annotators, yielding the *real-robot-grammar-generated* and *real-robot-human-annotated* test sets, respectively.

V. Results

A. Evaluation Metrics

Evaluating the performance of the EMV model is not trivial, e. g. as there are many ways to phrase the correct natural language answer to a given question and questions might also be ambiguous. To assess the quality of the model, we define the following output categories:

- *Correct*: Correct answer, no additional information.
- *TMI correct*: Correct answer with additional, correct information given ("tmi" = "too much information").
- *TMI wrong*: All information contained in the target utterance is given in the hypothesis, but some additional, wrong information is given, too.
- *Partially correct*: Some information contained in the target utterance is given in the hypothesis, some part is missing or wrong.
- *Partially correct, only action*: Every target fact is missing or wrong in the hypothesis, except for the action the robot talks about. This is a separate category because of the very limited number of actions in our simulated dataset.
- *Wrong*: Wrong answer given, but correct answer intent.
- *Inappropriate*: The answer is not even related to the question (e. g. "what did you do?" $\rightarrow$ "it is tuesday").

Furthermore, for each utterance, we count the number of facts contained therein. For example, the sentence "I moved the green cup to the sink" contains the three facts *move*, *green cup* and *sink*. By counting this contained information both in the model output as well as in the target utterance, we can calculate *information precision* and *recall*.

For the test datasets with grammar-generated dialog data, the structure and information content of possible questions and answers is known. Therefore, we developed a heuristic evaluation script, which categorizes a model response and calculates information precision and recall. The validity of this script was ensured by manually checking the script outputs for all possible types of grammar-generated questions (which is possible, since the grammar is known).

However, for the human-annotated QA data, the known question structure cannot be assumed anymore. Therefore, we do not use the scripted evaluation, but report human assessment (using the same categories as mentioned above) for these test sets.

[2]https://em.dataforlearningmachines.com/emv-data-collection/introduction

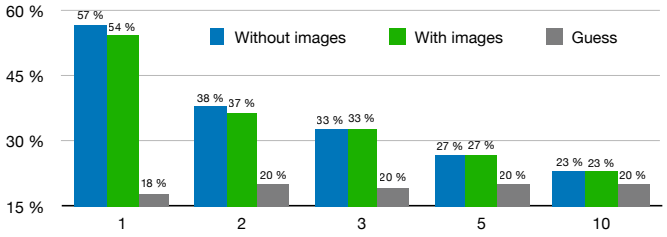

Fig. 3. Percentage of correct answers by history length.

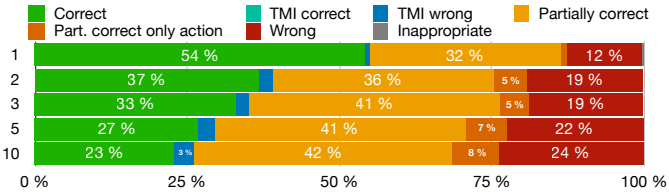

Fig. 4. Classification of answers by history length.

B. Experimental Results

To test generalization capability of our model to histories of different length, we evaluated on *simulated-robot-grammar-generated* test sets with histories of different numbers of episodes, including histories of length 10, i. e. double of the maximum length seen during training. The result of these assessments is presented in Fig. 3, which shows the percentage of correct answers over the length of the history. The EMV models trained and evaluated on episode data with or without images, respectively, are compared to the *guess* baseline model, which is the same model trained without using episode data at all, i. e. it receives only the question and needs to guess the answer. With this strategy, we account for the effect of the generated dataset, which is highly regular, therefore making the performance of text-only guessing quite high (about 20 % of correct answers). The results show the EMV models receiving episode data significantly outperform the guessing baseline. However, performance rapidly decreases with increasing history length, indicating the ability to resolve date-time references correctly and look up the correct entry in EM is very hard to learn. For a discussion of the with/without images performance in Fig. 3, see section V-C.

Fig. 4 shows a more detailed view on the answer classification results of the model with images after evaluation on histories of different lengths on the *simulated-robot-grammar-generated* test set. While the performance decline with increased history length is still evident, the graph additionally demonstrates that about three quarters of answers are at least partially correct, with only very few inappropriate answers at all. To compare, the *guess* baseline model reaches a performance of about 20 % correct, 4 % TMI wrong, 52 % partially correct and 20 % of wrong answers, independent of the history length. It gives nearly no inappropriate answer (< 0.1 %).

When taking a look at Fig. 5 showing information precision and recall as defined above, further observations can be made: Compared to the *guess* baseline, all EMV models have a better precision. However, the recall of guessing is very high,

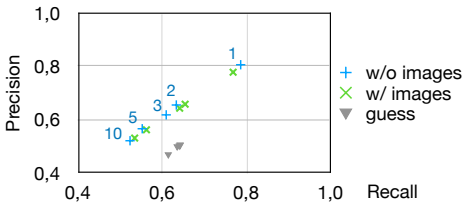

Fig. 5. Precision-Recall curve, referring to the information content of the utterances, as defined in section V-A. The different data points per model are the evaluation on histories of different length, as indicated by the numbers on the "w/o images" points.

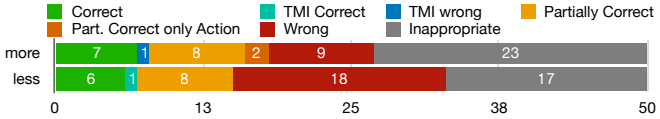

Fig. 6. Human evaluation results on the test set with human-annotated questions (absolute numbers). The two lines differ with respect to the number of trained parameters in the T5 speech network, with the lower one having less trained parameters.

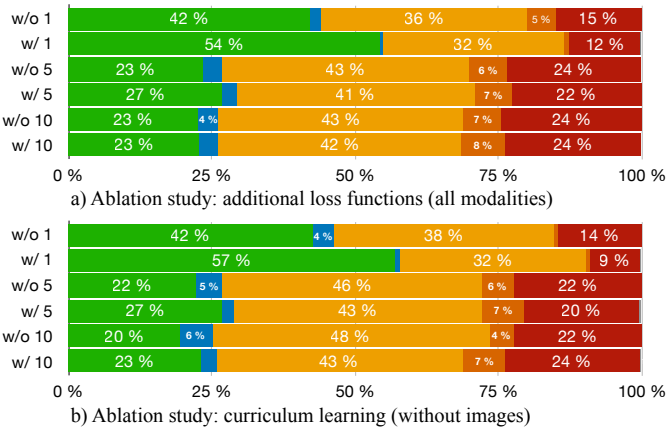

Fig. 7. Results of different ablation experiments. Each experiment compares the answer classification results on the *simulated-robot-grammar-generated* test set between a model without (w/o) and with (w/) the corresponding option. The numbers for each row (1, 5, 10) indicate the history length used for evaluation. This graph shares its legend with Fig. 6.

which is explained by the uninformed training favoring longer outputs containing many facts. Indeed, an analysis of the *guess* outputs shows that 59 % of all answers are equal to "I tried to but failed to move the multivitaminjuice and the redcup to the countertop", thus leading to a high recall but low precision. The episode-informed EMV models have no such overfitted peak answer.

To avoid the limitations of evaluation on the grammar-generated Q&A data, performance was also measured on simulated episodes annotated by human questions, as explained in section IV. The result of (manual) answer categorization on this generalization test set is shown in Fig. 6. Concerning the performance on human question, we can observe that the common language capabilities (which in theory should come from the pre-trained transformer network) are mostly lost and the model is not able to generalize well to unseen utterances. We additionally did tests on real robot recordings (also see the accompanying video). Here, the results are twofold: On the *real-robot-grammar-generated* set, answers are acceptable with a percentage of 39.3 % correct, 29.9 % partially correct, 28.7 % wrong and 0.4 % inappropriate answers on episodes of length one. For the *real-robot-human-annotated* set, despite its very small sample size, we can observe a similar result as for the *simulated-robot-human-annotated* data, with 2 correct, 1 TMI but correct, 1 wrong and 9 inappropriate answers. To conclude, the model mostly fails to transfer to new natural queries, while using real instead of simulated episodes has no such impact on performance.

C. Ablation Study

To further analyze the model, we performed additional ablation experiments. Fig. 3 already shows a comparison of a model trained including episode image data and one trained without these. The numbers show that adding the visual modality is not beneficial for performance, and for some history lengths even hurts. This can be explained by the generated dialog training data, which cannot leverage the information content of the images, therefore causing the model to ignore this modality.

While, due to the limitations by the generated train data, the multi-modal data cannot be properly exploited, we additionally analyzed other aspects of the EMV model training strategy. The first question is whether the complex loss function with its multiple contributions is actually useful. To investigate this, we compared training the EMV model (receiving all episode modalities) once with the complete loss function $\mathcal{L}$ as described in III-C, and once using only $\mathcal{L}_{NLG}$ (i. e. setting $\alpha_2 = \alpha_3 = 0$). Experimental results (see Fig. 7a) show that the additional loss functions provide a benefit concerning the percentage of correct answers on the evaluation on histories of up to length five (which is the maximum history length seen during training). However, this benefit vanishes when looking at the generalization to histories of length ten.

Another design aspect of the EMV model training is the use of curriculum learning with respect to the number of episodes in one history. Fig. 7b shows a comparison of two models, one trained directly on a dataset with histories of up to five episodes, and one with curriculum learning as described above, both excluding the episode modality of images. The results show that the proposed strategy is indeed useful, improving both the performance on histories of known length, as well as generalization to longer histories.

Concerning language generalization capability, we also experimented with a reduced number of trained parameters in the T5 speech encoder part of the model. While the results on the *simulated-robot-grammar-generated* set show no improvement, a slight improvement on unseen utterances in the *simulated-robot-human-annotated* test set can be observed. However, this effect (as shown in Fig. 6) is barely limited to moving some of the "inappropriate" answers to be categorized as "wrong".

VI. Conclusion & Discussion

We presented a deep-learning-based system for verbalization of a robot's experience encoded in an episodic memory. First, a dataset with episode recordings from simulation and real robot executions as well as question-answer pairs both grammar-generated and annotated by humans was collected. Second, we introduced the Episodic Memory Verbalization neural architecture, consisting of a speech encoder and decoder based on the Transformer network architecture, as well as an LSTM-based episodic memory encoder and some supplementary loss modules. Our experimental results show that the model is able to use content from episodic memory to answer natural language questions about past events and activities, but this ability shrinks with the amount of episodes in the memory.

While the presented results are promising, they can only be seen as a first step into the field of episodic-memory-based verbalization of robot experience. As outlined in section II, we are aware of other approaches that could solve the given problem, e.g. a symbolic episode database with rule-based verbalization. However, we chose to explore a deep learning approach for the following reasons: 1) When more human-annotated training data becomes available, we hope for a better generalization to variations in natural language, especially with the use of pre-trained language models. Hence, we plan to perform a large-scale crowd-sourced question and answer data collection to gather more realistic conversation data. 2) Using neural models enables a straight-forward combination of the highly multi-modal stream of episodic data and therefore does not require manually grounding symbolic information on subsymbolic recordings. 3) Future extension to other tasks requiring creation of and access to an EM might benefit from our findings, reuse the general architecture, or even be trained into the same model. Eventually, our vision is a model able to learn a full-featured, multi-functional EM integrated into a broader cognitive architecture.

References

[1] S. Rosenthal, S. P. Selvaraj, and M. Veloso, "Verbalization: Narration of autonomous robot experience," in *Int. Joint Conf. Artif. Intel.*, 2016, pp. 862–868.

[2] E. Tulving, "Episodic and semantic memory," *Organization of memory*, vol. 1, pp. 381–403, 1972.

[3] J. Rothfuss, F. Ferreira, E. E. Aksoy, Y. Zhou, and T. Asfour, "Deep episodic memory: Encoding, recalling, and predicting episodic experiences for robot action execution," *IEEE Robot. Automat. Let.*, vol. 3, no. 4, pp. 4007–4014, 2018.

[4] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2006, pp. 169–175.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Int. Conf. Neural Inf. Process. Syst.*, 2017, p. 11.

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[9] V. Perera, S. P. Selveraj, S. Rosenthal, and M. Veloso, "Dynamic generation and refinement of robot verbalization," in *IEEE Int. Symp. Rob. Hum. Interact. Comm.*, 2016, pp. 212–218.

[10] Q. Zhu, V. Perera, M. Wächter, T. Asfour, and M. M. Veloso, "Autonomous narration of humanoid robot kitchen task experience," in *IEEE-RAS Int. Conf. Humanoid Robots*, 2017, pp. 390–397.

[11] J. Winkler, M. Tenorth, A. K. Bozcuoglu, and M. Beetz, "CRAMm – memories for robots performing everyday manipulation activities," in *Adv. Cogn. Syst.*, vol. 3, 2014, pp. 47–66.

[12] M. Beetz, D. Bessler, A. Haidu, M. Pomarlan, A. K. Bozcuoglu, and G. Bartels, "Knowrob 2.0 — a 2nd generation knowledge processing framework for cognition-enabled robotic agents," in *IEEE Int. Conf. Robot. Automat.*, 2018.

[13] F. K. Kenfack, F. A. Siddiky, F. Balint-Benczedi, and M. Beetz, "RobotVQA - a scene-graph- and deep-learning-based visual question answering system for robot manipulation," in *IEEE/RSJ Int. Conf. Intel. Rob. Syst.*, 2020.

[14] G.-M. Park, Y.-H. Yoo, D.-H. Kim, and J.-H. Kim, "Deep ART neural model for biologically inspired episodic memory and its application to task performance of robots," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1786–1799, 2018.

[15] W. Wang, B. Subagdja, A. Tan, and J. A. Starzyk, "Neural modeling of episodic memory: Encoding, retrieval, and forgetting," *IEEE Trans. Neur. Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1574–1586, 2012.

[16] P.-H. Chang and A.-H. Tan, "Encoding and recall of spatio-temporal episodic memory in real time," in *Int. Joint Conf. Artif. Intel.*, 2017, pp. 1490–1496.

[17] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "Temporal cycle-consistency learning," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019.

[18] J. Lei, L. Yu, T. L. Berg, and M. Bansal, "TVQA+: Spatio-temporal grounding for video question answering," *arXiv:1904.11574*, 2019.

[19] Z. Yu, D. Xu, J. Yu, T. Yu, Z. Zhao, Y. Zhuang, and D. Tao, "ActivityNet-QA: A dataset for understanding complex web videos via question answering," *AAAI*, vol. 33, no. 1, pp. 9127–9134, 2019.

[20] Z. Zhao, S. Xiao, Z. Song, C. Lu, J. Xiao, and Y. Zhuang, "Open-ended video question answering via multi-modal conditional adversarial networks," *IEEE Trans. on Image Process.*, vol. 29, pp. 3859–3870, 2020.

[21] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 1378–1387.

[22] C.-S. Wu, R. Socher, and C. Xiong, "Global-to-local memory pointer networks for task-oriented dialogue," in *Int. Conf. Learn. Representations*, 2019.

[23] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Int. Conf. Learn. Representations*, 2015.

[24] T. Kollar, S. Tellex, M. Walter, A. Huang, A. Bachrach, S. Hemachandra, E. Brunskill, A. Banerjee, D. Roy, S. Teller, and N. Roy, "Generalized grounding graphs: A probabilistic framework for understanding grounded commands," *arXiv:1712.01097*, 2017.

[25] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Grounding verbs of motion in natural language commands to robots," in *Experimental Robotics*, vol. 79, Springer Berlin Heidelberg, 2014, pp. 31–47.

[26] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2d pose estimation using part affinity fields," in *arXiv:1812.08008*, 2018.

[27] N. Vahrenkamp, M. Wächter, M. Kröhnert, K. Welke, and T. Asfour, "The robot software framework ArmarX," *it - Information Technology*, vol. 57, 2015.

[28] A. Bordes, Y.-L. Boureau, and J. Weston, "Learning end-to-end goal-oriented dialog," in *Int. Conf. Learn. Representations*, 2017.