Speech interaction strategies for a humanoid assistant

Sebastian Stüker*, *Stefan* Constantin, *Jan* Niehues, *Thai-Son* Nguyen, *Markus* Müller, *Ngoc Quan* Pham, *Robin* Rüde and *Alex* Waibel
Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract. The goal of SecondHands, a H2020 project, is to design a robot that can offer help to a maintenance technician in a proactive manner. The robot is to act as a second pair of hands that can assist the technician when he is in need of help. In order for the robot to be of real help to the technician, it needs to understand his needs and follow his commands. Interaction via speech is a crucial part of this. Due to the nature of the situation in which the interactions take place, often the technician needs to speak to the robot when under stress performing strenuous physical labor, the classical turn based interaction schemes need to be transformed into dialogue systems that perform stream processing, anticipating user intentions, correcting itself as more information become available, in order to be able to respond in a rapid manner. In order to meet these demands, we are developing low-latency streaming based automatic speech recognition systems in combination with recurrent neural network based Natural Language Understanding systems that perform slot filling and intent recognition in order for the robot to provide assistance in a rapid manner, that can be partly based on speculative classifications that are then being refined as more speech becomes available.

1 Introduction

The goal of the project SecondHands, sponsored within the European Union's 8th framework programme *Horizon 2020*, is to develop a humanoid robot that acts as an assistant to maintenance technicians. It is supposed to either proactively support the technician in performing routine maintenance operations, or to offer support upon request from the human. Within the project a consortium of five partners are collaborating on this task: Ocado, Karlsruhe Institute of Technology (KIT), École Polytechnique Fédérale de Lausanne (EPFL), Ocado, and Universitá di Roma Sapienza.

Conceptually, the robot's task is to provide a second pair of hands to the maintenance technician, such that once the robot has been trained, it can predict when it can usefully provide help and knows which actions to take to provide it. In order to reach this goal the project aims to advance abilities and key technologies relevant to industrial robotics, such as cognition, human-robot interaction, mechatronics, and perception.

To operate within environments designed primarily for industrial efficiency in cooperation with a mostly human workforce, a robot needs a rich repertoire of human-like skills, and, in the opinion of the project's partners a humanoid or human-like form, specifically in order to use the same methods of access and to be able to efficiently cooperate with humans when they are performing their tasks. One key perceptual and cognitive capability in this cooperation is the communication with the human co-workers via natural language. Traditionally this means the construction of a spoken dialog system that consists of an automatic speech recognition system, a natural language understanding system, a dialog manger, a natural language generation system and a speech synthesis system. The construction of goal oriented dialogues in this manner often leads to heavily turn-based systems that require several turns to resolve ambiguities and correct errors in an explicit manner, in order to finally reach the goal.

However this approach is ill-suited in the human assistance scenario, as the human does not have the time nor the cognitive capacities to conduct this kind of dialogue while performing his strenuous task.

KIT has therefore set-up to develop a slightly different kind of dialogue system that works in a stream-based manner. That is to process the human speech in a run-on manner to feed this continuous low-latency stream of words from it into a stream-based natural language understanding component that passes information on to the planner of the robot as it becomes available, correcting itself if necessary, when more information has become available.

Also, the dialogue is supposed to be become more "social" in order to improve the acceptance of the robot by the human technicians.

In addition we are pursuing novel, alternative approaches for end-to-end dialogue modelling that are based on memory-networks, and which in their attributes seem to be specifically suited for the use case at hand.

The dialog systems constructed in this manner have become part of the robot that has been constructed as a proof-of-concept within the project: ARMAR VI. It will be verified against a series of real maintenance tasks that

*e-mail: sebastian.stueker@kit.edu

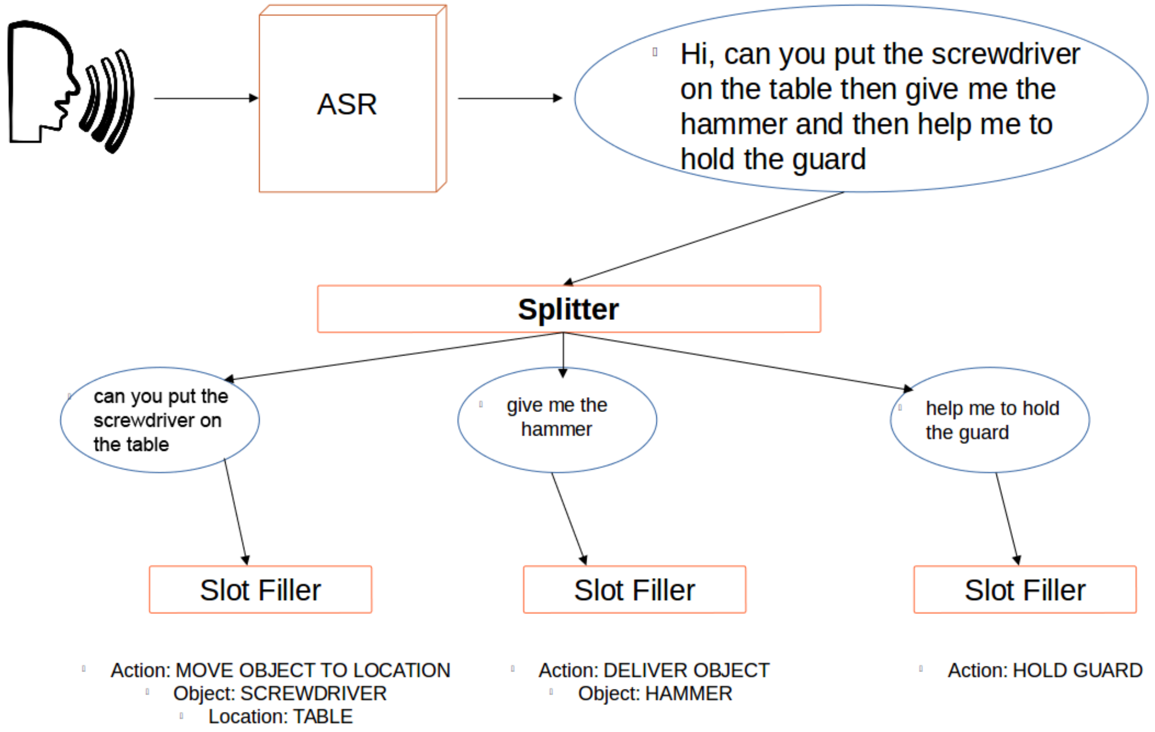
ASR
Hi, can you put the screwdriver on the table then give me the hammer and then help me to hold the guard
Splitter
can you put the screwdriver on the table
give me the hammer
help me to hold the guard
Slot Filler
Slot Filler
Slot Filler
Action: MOVE OBJECT TO LOCATION
Object: SCREWDRIVER
Location: TABLE
Action: DELIVER OBJECT
Object: HAMMER
Action: HOLD GUARD

Figure 1. Natural language understanding pipeline.

occur within the Ocado's highly automated Customer Fulfilment Centres (CFCs).

2 Low latency speech recognition

Currently, in the design of modern online automatic speech recognition (ASR) systems, there is usually an audio segmenter component which seeks to detect complete segments in input audio streams, often based on detected silence. Later, an ASR component is in-charge of decoding the detected segments into text. In order for the system to keep up with the audio stream it has to run in real-time or faster. The faster the decoding runs the lower the latency which is bounded by the length of the detected segments.

In our system [1], the ASR component was built to process an audio stream and output a stream of transcripts which is the input to the subsequent components (e.g., Natural Language Processing). By using a dynamic decoding framework for speech recognition, we can avoid the detection of audio segments, and directly perform the decoding as soon as a small part of speech is recorded. Specifically, the decoding framework allows us to compute acoustic feature vectors every 0.25 seconds of the speech data and dynamically extend the search network by processing the new frames. This so called run-on recognition helps us to avoid the latency caused by waiting for the end of the current segment.

Normally, only at the end of an utterance the most probable hypothesis is determined. However, since waiting until the end of the utterance leads to a long latency, we detect when a part of the hypothesis becomes stable and can be kept. This is possible due to the fact that our search implements a viterbi beam search, so that for earlier portions of a hypothesis it is possible and frequently happens, that all competing alternatives get pruned away later in the search.

We also introduced a novel protocol in which the ASR component and the subsequent components follow to dramatically reduces the latency while still maintain the recognition accuracy. Basically, we output probable parts of the unstable hypothesis and send them to the subsequent component. Later, the recognizer can revise its decision and overwrite the previous output if necessary. In this way, the recognition component does not need to wait until a stable portion or end-of-segment, instead it finds the most probable hypothesis every iteration of the incremental decoding, detects and sends the update portions to the subsequent component.

3 Natural language understanding in stream mode

We designed a component that takes the output of the ASR counterpart and extract the information required from the utterance. Specifically, the main task of the Natural Language Understanding (NLU) component is to identify one of the predefined actions, and the corresponding objects and locations. The actions are revolved around transporting the items from and to the technicians, as well as helping the technicians in handling the guards. The included objects are the tools required in the working environment, such as screwdriver, hammer, and ratchet-spanner. Our pipeline is illustrated in Figure 1.

We use a standard recurrent neural network with Long-Short Term Memory (LSTM) units [2] to encode the whole utterance. The main NLU model is showed in Figure 2. The network runs in a bidirectional manner to get the contextual information from both sides of each word [3]. For predicting the objects and the locations, we used the local context which are the states at each time step. However, for predicting the actions, we found that it is beneficial to utilize the global context by joining the states of the first and the last steps. We also design our system to be able to detect multiple actions in one utterance. In order to solve this problem, we train an addition sentence splitter which is also fundamentally a bidirectional recurrent neural network. The sentence splitter annotates the splitting point of the long sentence. For example, "can you put the screwdriver on the table then give me the hammer and then help me to hold the guard" is split into "can you put the screwdriver on the table", "give me the hammer", and "help me to hold the guard". We generate the training data for this splitter simply by randomly concatenating sentences in our training data.

Our NLU component currently performs recognition whenever the segment is sent by the ASR component. The segment splitting of the stream is considered a subtask in the ASR component. Potentially, our component can also be minimally modified to be able to process the stream in a token basis. Such design allows us to be able to produce the prediction sooner than waiting for the whole utterance to finish.

4 End-to-end dialog modelling with memory networks

The NLU component, described in section 3, process the splitted sentence parts individually without regarding previous parts or asking questions. However, in the human-robot interaction, it is often necessary to regard the previous parts of a dialog. For example, the human asks the robot to redo the last action for another object. Often, it is also necessary to have multiple interactions between the human and the robot before an action can be executed by the robot. For example, a human asks a robot to hand over a screwdriver. In this example, the information, what size the screwdriver should have, is missing. Therefore, a robot should be able to ask for this information and after getting the information, it can hand over the user a suitable screwdriver. The robot should also be able to ask questions to resolve ambiguities. To be able to regard previous parts of the dialog and ask questions, a Dialog Manager (DM) and a Natural Language Generation (NLG) component are necessary. A dialog system with a DM, NLU, and NLG component that can be trained end-to-end make it possible to change or extend the domain of the dialog system without changing the dialog system. It is only necessary to train the system with an extended or changed dataset.

The end-to-end trainable Memory Network [4] is an interesting neural network architecture for using it as NLU, NLG, and DM component that can be trained end-to-end [5]. It is a neural network architecture that has access to the memory. All utterances before the current utterance of the user are saved in the memory as utterance-embeddings. To every utterance-embedding in the memory, it is calculated how relevant it is to the last utterance of the user. All memory entries are multiplied by their relevance and summed up. This sum plus the utterance-embedding of the last utterance of the user is the output of the Memory Network. This conforms to the NLU and DM component of a dialog system. The output of the Memory Network is multiplied by a trainable matrix. Each vector representation of a possible output candidate is multiplicated by that result. The candidate with the highest value is the output of the system. This conforms to the NLG component of a dialog system. An alternative to this kind of NLG is to generate the answers word by word [6]. The trainable matrices are trained by the backpropagation algorithm.

The Memory Network achieved good results in experiments with the Dialog bAbI Tasks. In the first task of the Dialog bAbI Tasks a dialog system has to request for four restaurant reservation parameters and should be able to generate after receiving all parameters an API call with them. The dialog system has achieved more than 99 % accuracy in the first task of the Dialog bAbI Tasks. To test if the system is suitable for the robot domain, we made a dataset similar to the bAbI Dialog tasks for the robot domain. In this dataset, the dialog system achieved an accuracy of more than 95 %.

5 Non-verbal cues as feedback for the user

Backchannel (BC) responses like "uh-huh", "yeah", "right" are used by the listener in a social dialog as a way to provide feedback to the speaker. In the context of human-computer interaction, these responses can be used by an artificial agent to build rapport in conversations with users and make the human-robot interaction more social. In the past, we proposed an approach by training artificial neural networks on acoustic features such as pitch and power and also attempted to add word embeddings via word2vec [7]. We now refined this approach by evaluating different methods to add timed word embeddings via word2vec. Comparing the performance using various feature combinations, we observed that adding linguistic features improved the performance over a prediction system that only uses acoustic features [8]. The most commonly used acoustic features in related research are fast and slow voice pitch slopes and pauses of varying lengths. A neural network is able to learn advantageous feature representations on its own. By inputting the absolute pitch and power (signal energy) values for a given time context, the network is able to automatically extract the pitch slopes and pause triggers by subtracting the neighboring values in the time context for each feature. We train the network on the outputs [1, 0] for BCs and [0, 1] for non-BCs. The placement of BCs is dependent on previous BCs: If the previous BC utterance was a while ago, the probability of a BC happening shortly is higher and vice versa. After each BC, the

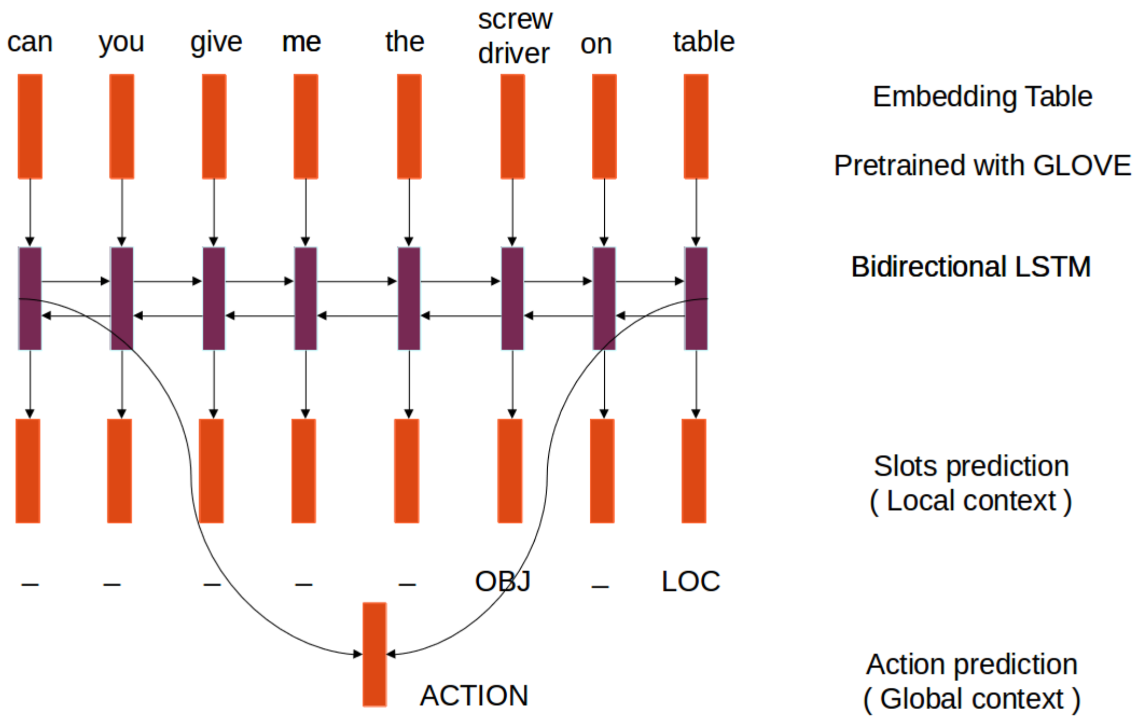

Figure 2. Natural language understanding model details.

probability of a new BC rises over time. To accommodate for this, we want the neural network to also take its previous internal state or outputs into account. We do this by using LSTMs instead of dense feed forward layers. Our goal is to generate an artificial audio track containing utterances such as “uh-huh” or “yeah” at appropriate times. The neural network outputs a noisy value between 0 and 1. To generate an audio track from this output, we need to convert this noisy floating probability value into discrete trigger time stamps: We first run a low-pass filter over the network output, which removes all the higher frequency components and yields to a less noisy and more continuous output function. To ensure our predictor does not use any future information, this low-pass filter must be causal. The commonly used gaussian filter is symmetric, which in our case means it uses future information as well as past information. To prevent this, we cut the right side of the filter off asymmetrically at some multiple c of the standard deviation σ. Then we shift the filter to the left so the last frame it uses is ± from the prediction target time. This means the latency of our prediction increases by $c \cdot \sigma ms$. If we choose $c = 0$, we cut off the complete right half of the bell curve, meaning we do not need to shift the filter, which keeps the latency at 0 ms, but at the cost of accuracy of the low-pass filter.

All of the results in Table 1 use the following setup: two LSTM layers (with 70 and 35 neurons), a context width of 1500 ms, a context frame stride of 2, and the margin of error is 0ms to +1000 ms. We initialized the weights using Glorot uniform initialization and used Adam for optimization. These parameters were chosen to give the best results on the validation data set. A more detailed comparison of training methods, feature combinations, context widths and network layouts can be seen in our previous work [7]. The experimental setup is slightly different there, so the absolute F1-values should not be directly compared. Precision, recall and F1-Score are given for the completely independent evaluation data set. The performance with both linguistic and acoustic features is significantly better than with just acoustic features ($p < 1$ %). The improvement of adding the ICSI meeting corpus to word2vec training is not statistically significant.

6 Conclusion

In this paper we have given an insight into the dialogue modelling activities that we are performing within the H2020 project *SecondHands*. The goal of the project is the construction of a humanoid robot that acts as an assistant to the human technicians performing maintenance tasks.

In order for the robot to collaborate well with the technician, we have shown several ways for building a stream-based spoken dialogue system that offers low-latency and does not rely on turns. We thereby made use of stream-based speech recognition that works with speculative output, of recurrent-neural-network based natural language understanding that can operate in a stream-based manner, or of end-to-end dialogsystems based on memory networks.

In order for the dialog to be more socially acceptive to the human user, we also demonstrated a non-verbal backchannel system that can support the communication with the user.

Table 1. Objective results with various input feature combinations.

Features	Precision	Recall	F1-Score
power only	0.244	0.516	0.331
accoustic (power, pitch, ffv)	0.279	0.515	0.362
linguistic (word2vec$_{dim=30}$)	0.244	0.478	0.323
accoustic and linguistic (power, pitch, ffv, word2vec$_{dim=30}$):			
· word2vec trained on Switchboard	0.298	0.510	0.376
· word2vec trained on Switchboard	0.305	0.519	0.385

This work has been conducted in the SecondHands project which has received funding from the European Union's Horizon 2020 Research and Innovation programme (call:H2020- ICT-2014-1, RIA) under grant agreement No 643950.

References

[1] J. Niehues, T.S. Nguyen, E. Cho, T.L. Ha, K. Kilgour, M. Müller, M. Sperber, S. Stüker, A. Waibel, Interspeech 2016 2513–2517 (2016)

[2] S. Hochreiter, J. Schmidhuber, Neural computation **9**, 1735 (1997)

[3] G. Mesnil, X. He, L. Deng, Y. Bengio, *Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding.*, in *Interspeech* (2013), 3771–3775

[4] S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus, *End-To-End Memory Networks* (2015), 2440–2448, `http://papers.nips.cc/paper/5846-end-to-end-memory-networks`

[5] A. Bordes, Y.L. Boureau, J. Weston, *Learning End-to-End Goal-Oriented Dialog*, in *Proceedings of the International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017* (2017), `http://arxiv.org/abs/1605.07683v4`

[6] S. Constantin, J. Niehues, A. Waibel, *An End-to-End Goal-Oriented Dialog System with a Generative Natural Language Response Generation*, in *Proceedings of the International Workshop on Spoken Dialogue Systems, IWSDS 2018, Singapore, Singapore, May 14-16, 2018 (forthcoming)* (2018)

[7] R. Ruede, M. Müller, S. Stüker, A. Waibel, *Yeah, Right, Uh-Huh: A Deep Learning Backchannel Predictor*, in *Proceedings of the Eight International Workshop on Spoken Dialogue System (IWSDS 2017)* (Pittsburgh PA, U.S., 2017)

[8] R. Ruede, M. Müller, S. Stüker, A. Waibel, *Enhancing Backchannel Prediction Using Word Embeddings*, in *Proceedings of INTERSPEECH* (Stockholm, Sweden, 2017)