UvA@Home Team Description Paper 2017

Jonathan Gerbscheid, Thomas Groot, and Arnoud Visser

University of Amsterdam
Faculty of Science
The Netherlands
http://www.uvahome.nl/

Abstract. This team description paper describes the approaches that will be taken by the UvA@Home team to compete in Standard Platform League with the Softbank Robotics Pepper. The research challenges concern person recognition, natural language processing and navigation. Modules implemented so far include people detection, speech recognition and natural language processing. The remaining challenges will be solved using the previous research and achievements of the UvA teams in the RoboCup.

1 Introduction

The UvA@Home team consists of two bachelor Artificial Intelligence students supported by a senior university staff member. The team was founded as a part of the Intelligent Robotics Lab (IRL) at the beginning of the 2016-2017 academic year. The IRL acts as a governing body for all the University of Amsterdam's robotics teams, including the Dutch NAO Team and the UvA@Home team (both active in a RoboCup Standard Platform League). It encourages the sharing of experience between these teams to be successful in both leagues, which is possible because the Nao and the Pepper robot share the same NaoQi basis (although a slightly different version).

2 Background

The Universiteit van Amsterdam has a very long history in RoboCup [1]. The focus of the research of the university is on perception, world modeling and decision making. The @Home competition nicely fits in our research; the lack of a standard platform withheld us from entering the competition. Instead, we have initiated studies towards the simulation of the @Home competition [2, 3].

After the qualification for this league, the Intelligent Robotics Labhas bought a Pepper robot under the conditions of Softbank Robotics. In addition, the university has good contact with two Dutch companies in the possession of a Pepper robot.

3 Challenge

The Social Standard Platform League (SSPL) imposed a new challenge inside the RoboCup @Home competition. The idea behind the RoboCup@Home is to shown the performance of robots executing domestic tasks [4]. For the SSPL, the focus will be on a robot who will actively look for interaction with humans. Hence, this league focuses on Natural Language Processing, People Detection and Recognition, and Reactive Behaviors. To demonstrate this skills, a cocktail party scenario is invented as challenge [5]. Progress in this league will be directly applicable to social relevant scenarios and can directly be disseminated to interested companies and the community.

4 Scientific contribution

4.1 Dialog model

For a social challenge a natural, robot-led, human-robot interaction is important. In the @Home competition a robot has to discover what drinks a customer wants, which is made possible by a combination of speech recognition, understanding and generation [6]. Speech was recognized using Google Cloud's speech to text API, understood by matching either the object or main verb of a sentence against a list of key words and, finally, generated using templates with variable parts. The difficulty lies in the large quantity of key words, as they are based on the properties of the ordered drinks. The obtained precision when identifying the unavailable drinks was 0.625 and the obtained recall was 1.0, resulting in an F_1 measure of 0.769.

The first step towards this result is to understand order of a customer. Understanding natural language requires as first step a written format of the sentence that was spoken, which was obtained using the Google Cloud speech to text API[1], which takes an audio file as input and returns a transcription of the spoken sentences in the audio file as output using CLDNN-HMM, which combines a Convolutional Neural Network (CNN), a Deep Neural Network (DNN), and Long Short-Term Memory (LSTM) [7].

Naturally, the robot needs to know when the customer is speaking and, therefore, the customer can indicate that he wants to speak by touching and holding the back of the robot's left hand, which is similar to the method that was used by [8], as they also used the robot's hand sensor to determine when to start listening. However, the difference is that [8] used signal energy to determine when to stop listening, while the robot in this research stopped listening once the customer let go of the robot's hand. In order to indicate to the customer when the robot was listening, the blue LEDs in its eyes would rotate. Furthermore, the audio file that was recorded during that time was automatically saved on the robot and sent to be transcribed by the Google Cloud speech to text API.

[1] `https://cloud.google.com/speech/`

There are several steps that were taken in order to understand the content of a written sentence. As first step, the sentence was parsed using the Stanford Dependency Parser[2], and the main verb was extracted using the parsed sentence and NLTK's[3] `pos_tag` method [9], which processes a sequence of words and attaches a part-of-speech tag to each word.

During the second step, the type of the given answer was analyzed. The types were categorized into 'empty' and 'non-empty' answers: an 'empty' answer is an answer such as "Yes" or "No, I don't", while a 'non-empty' answer is an answer such as "I don't have any lemons". The module used the main verb, object and, optionally, the negation to understand a written sentence. If the customer gave an empty answer, then the main verb and object of the question were used to understand the sentence instead of those of the answer. However, the negation of the answer was always used.

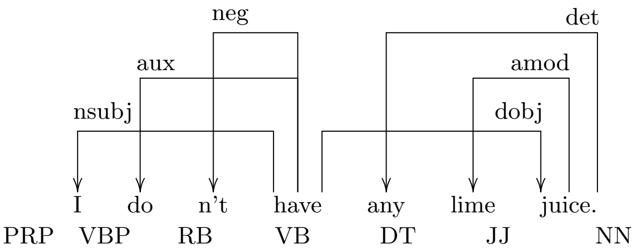

Fig. 1. A visualisation of the output of the Stanford Dependency Parser.

The third step was to analyse the sentence itself, which required identifying the object of the sentence and, optionally, the negation. The parser labelled the object as 'dobj' and the negation as 'neg', as can be seen in Figure 1. Each found object was added to a list of objects and, similarly, if a negation was detected then 'not' was added to a list of negations. However, if no negation occurred in the written sentence, 'None' was added to the list of negations instead.

Using the obtained features, namely the verb, object and negation, the program could understand the sentence by matching either the main verb or object against a list of key words, which depended on whether the main verb was 'possessive', e.g. if it was a verb such as 'have' or 'own'. If it was, the object was matched against the key words, while the main verb was matched if it was not. If no match was found, the robot did not understand what the customer said. However, if a match was found, then the robot could update the list of available drink properties or remove a drink from the list of available drinks.

[2] https://nlp.stanford.edu/software/lex-parser.shtml

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

4.2 People Detection

People detection is a critical part of human-robot interaction and advancements in people detection will improve the level of interaction that can be achieved [10]. It has been approached using different sensors and techniques, however the capabilities of detection with the Pepper have not yet been examined properly. Detection techniques using the different sensors available to the Pepper are explored and a state of the art convolutional neural network and 3D blob detector are developed . The detectors are then combined using a detection history based approach. Results show that performance of the CNN, although high for cases with 1-3 test subjects, decreases significantly in crowded settings. The addition of 3D data to reuse previous detections was shown to increase recall, however due to the limited range of the 3D sensor, recall remained lower than that achieved on the lower person count test cases.

The network that is trained is an Inception_v1 network [11] provided by the Tensoflow framework [12]. This network is then trained on a dataset consisting of various people detection datasets from TU-Dresden [13][14].

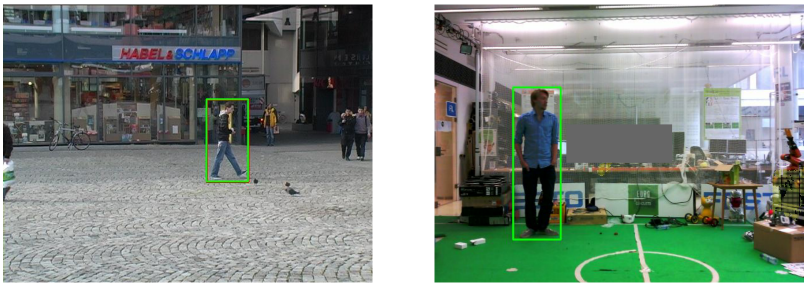

Fig. 2. People detection by CNN examples. Left: Training on TUD-Motionpairs, Right: Verification in lab (IRL dataset)

The people are detected using not only the 2D camera, but depth information of the 3D camera is used as well. The approach taken to 3D people detection is to find blobs in the image that are potentially people. This approach is not intended to function well as a single detector, but instead serves to solidify detections made by the 2D detector by making faster but less reliable detections by searching for areas of points the image that are similar in depth.

Human sized blobs are found by searching for areas with similar depth using a slightly modified version of the flood fill algorithm [15]. This search is initiated from starting points on a grid; because of the vertical shape of standing people the grid is more dense on the horizontal axis.

The final step in the combined detection algorithm is the combination of the two different detectors. The convolutional neural network has returned bounding boxes and confidences, while the depth image detector has returned blobs of indices. The first step in combining these two detections is to transform them to

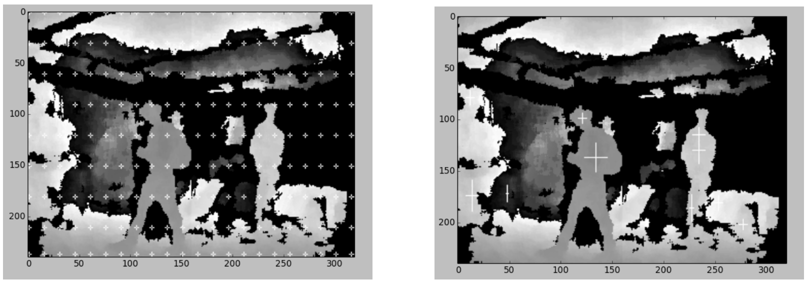

Fig. 3. 3D detection. Left: Image with grid, Right: Final Detections

the same coordinate system. Subsequently, all centroids that lie within of a CNN bounding box are selected. The percentage of indices of the blob that lie within the bounding box is then calculated, if this is higher than 50% the detection is accepted. This is the strictest method and while it removes up to 99% of all false positives, it also removes all detections where one of the two detectors did not correctly identify the person, significantly impacting recall.

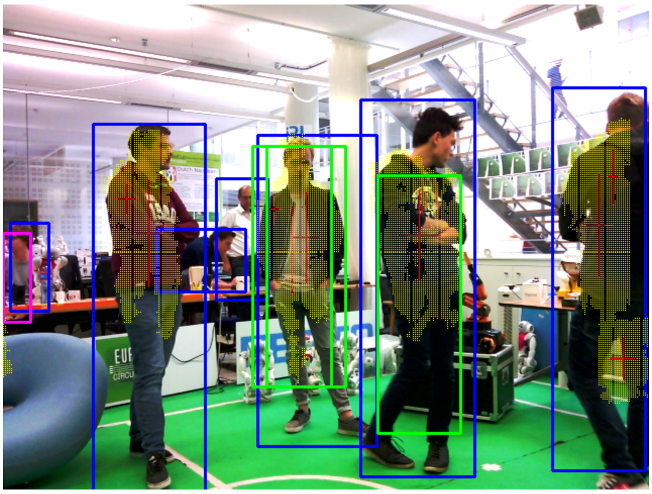

Fig. 4. An example where the CNN did not correctly detect people that were detected in the previous frame, but were then detected using the history approach. Ground truth in blue, CNN detections in purple, blob shapes in yellow, blob centroids in red and with detections obtained through the history with 3D blobs in green

So, instead of direct filtering, a detection history is used to stabilize detections. In the current frame all detections are accepted and are placed in a history. All blob locations that are not for at least 50% inside a CNN detection are then checked in the history. If a detection was made in the history at the same location, 50% overlap between the blob and bounding box, as the blob in the current frame, the detection from the history is reused. This stabilizes the detector by still correctly finding people in the frames where the CNN failed to do so of which the results can be seen in figure 4.

5 Open Challenge

The UvA@Home team created a system that is able to inform a user of news articles with an opinionated undertone [16]. To start interaction a person first stands in front of the robot. The system creates a user profile by recognizing the persons face through the OpenFace Deep Neural Net[17]. After having created a user profile the person can start telling the system its preferences. The speech recognition is done using the Google Speech Recognition Cloud API [4].

The person first tells the system his/her preferences which are then stored. During this process opinions on topics can be asked as well. News is scraped from a variety of popular news websites (Reuters, CNN, BBC) using Beautiful Soup [5]. The system can answer basic queries using a rule-based approach which are parsed using the *Standford POS tagger* [18]. The system uses the *Standford POS tagger* to turn sentences into syntax trees and parse the lowest laying *noun phrase* (NP) in the tree. Studying leafs of other NPs the system is able to derive meaning from questions given by the user. The conversation domain is generally limited, so only a few interpretations of sensible trees (that are relevant for the conversation) are possible. During the conversation the person can give feedback on specific queries. The system will remember this and will update the user profile so that the next answer to a query will be more relevant to the user.

During the conversation the person can also ask the system about its opinion on certain topics. The system will scan posts on Twitter [6] in order to gather a consensus regarding the topic. The topic will have to be frequently mentioned on the Twitter to give reliable output.

6 Conclusions and future work

We are looking forward to demonstrate our research for the Softbank Robotics Pepper robot and our progress on the challenges imposed by the RoboCup@Home competition. The current working modules have all been tested on the Nao robot and on the Pepper robot as they share the same operating system. A large benefit of this league is that the achievements made are directly applicable to relevant scenarios in a social environment, something that can directly be communicated and disseminated to interested companies and the community.

[4] https://cloud.google.com/speech/

[5] https://www.crummy.com/software/BeautifulSoup/

[6] https://twitter.com/

Bibliography

[1] Emiel Corten and Erik Rondema. Team description of the windmill wanderers. In *Proceedings on the second Robocup Workshop*, pages 347–352, July 1998.

[2] Sander van Noort and Arnoud Visser. *Extending Virtual Robots towards RoboCup Soccer Simulation and @Home*, pages 332–343. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[3] Victor I.C. Hofstede. The importance and purpose of simulation in robotics. Bachelor thesis, Universiteit van Amsterdam, June 2015.

[4] L. Iocchi, D. Holz, J. Ruiz-del Solar, K. Sugiura, and T. van der Zant. RoboCup@Home: Analysis and results of evolving competitions for domestic and service robots. *Artificial Intelligence*, 229:258–281, 2015.

[5] Arnoud Visser. A new robocup@ home challenge. *Benelux A.I. Newsletter*, 31(1):3–6, 2017.

[6] Tirza F.E. Soute. Discovering available drinks through natural, robot-led, human-robot interaction between a waiter and a bartender. Bachelor thesis, Universiteit van Amsterdam, July 2017.

[7] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *International Conference on Acoustics, Speech and Signal Processing*, page 2, 2016.

[8] Vittorio Perera, Tiago Pereira, Jonathan Connell, and Manuela M. Veloso. Setting up pepper for autonomous navigation and personalized interaction with users. *Computing Research Repository*, 1704, 2017.

[9] Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the International Committee on Computation Linguistics and Association for Computational Linguistics on Interactive Presentation Sessions*, International Committee on Computation Linguistics and Association for Computational Linguistics '06, pages 69–72, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[10] Jonathan R. Gerbscheid. People detection on the pepper robot using convolutional neural networks and 3d blob detection. Bachelor thesis, Universiteit van Amsterdam, July 2017.

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[12] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems. arXiv preprint arXiv:1603.04467, 2016. Software available from tensorflow.org.

[13] Christian Wojek, Stefan Walk, and Bernt Schiele. Multi-cue onboard pedestrian detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 794–801. IEEE, 2009.

[14] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021. IEEE, 2009.

[15] Theo Pavlidis. Filling algorithms for raster graphics. *Computer graphics and image processing*, 10(2):126–141, 1979.

[16] Jonathan Gerbscheid, Thomas Groot, Joram Wessels, Rijnder Wever, and Wijnand Van Woerkom. Personalized news conversations with the softbank pepper. project report, Universiteit van Amsterdam, March 2017.

[17] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[18] Kristina Toutanova and Christopher D Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics, 2000.

Pepper Robot's Hardware Description

This section covers the technical aspects of the Softbank Pepper robot that are relevant for this challenge. The Pepper robot is around 1.2 meters in height, see 5 and weighs 29kg. It is equipped with a microphone, two 2D cameras, a 3D Sensor, Laser Range Finders, Infrared Sensors and two ultrasonic sensors[2].

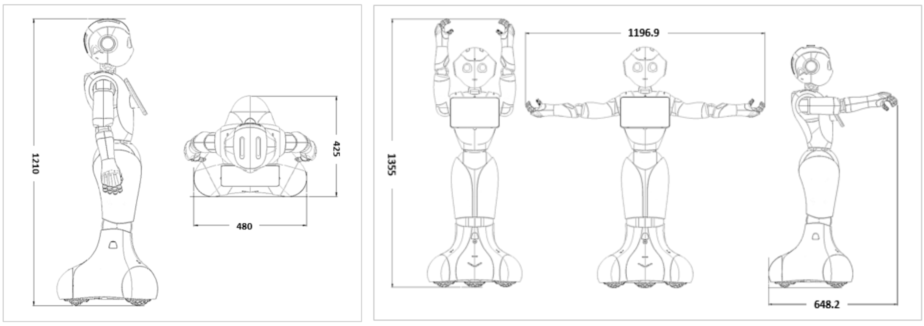

Fig. 5. Dimensions of the Pepper in mm[2]

6.1 2D cameras

The Pepper has two cameras that are located on the forehead and in the mouth of the robot. Both cameras have a horizontal field of view of 55.2°and a vertical field of view of 44.3 °, the fields of view of the two cameras intersect from 1̃00 cm.

6.2 3D sensor

The 3D Sensor used in the Pepper is a version of the Asus Xtion 3D sensor and is located behind the eyes of the Pepper. Its horizontal and vertical field of view is slightly larger than that of the 2D cameras and it is pointed in the same direction as the upper 2D camera.

Software List

Main software

- Operating System/Robot Control: Naoqi.
- Face recognition: OpenFace.
- Navigation: Both Naoqi/ROS based SLAM modules.
- Conversation: described in [6]

Used Cloud service:

- Speech recognition: Google Cloud's speech to text API.
- People detection: described in [10]