Development of a dialogue system for small talk using visual sensation.

M.G. Meedendorp[1]

Abstract. Conversational software is software with which the user can interact in a natural language such as Japanese or English. There are two main approaches to conversational software, chatbots and dialogue systems. Chatbots are simple systems which generate responses by matching keywords or patterns on the user input. Dialogue systems, on the other hand, interpret natural language and try to extract the meaning out of the sentences and keep track of the state of the conversation such as topics discussed or who spoke last. Both of these systems have advantages and disadvantages. Chatbots are easy to create and require no thorough understanding of the linguistics involved. The comparatively more advanced dialogue systems are a lot more human-like in conversation and easier to communicate with, but also require a lot of specialized knowledge to make.

The goal of this study is to improve on the chatbot concept and make its performance more like a dialogue system without the associated complexity. Such a system should be more versatile than a chatbot, by switching topics on its own and initiating conversations without user input. In this paper this is attempted by providing a chatbot-style system with visual input in addition to the natural language input. This system was implemented on a SoTa robot, with visual input consisting of different labels (e.g. 'car', 'cat', 'person') and location data for objects recognized from a camera by an object recognition and classification algorithm.

Our system with visual input was indeed able to initiate a conversation without input from the user and able to switch topics by utilizing the visual input. In addition it has major advantage over advanced dialogue systems because our system does not require a high level of linguistic knowledge to create. From these results we can conclude that the system is an improvement over chatbot systems limited to textual input and less complex than an advanced dialogue system. This makes the system is perfectly suited for situations where a more context-oriented conversation is desirable but a complex dialogue system is not necessary or feasible. The current implementation of the system is limited by the fact that it does not allow for any textual user input other than answering yes/no questions, so is not able to show the full potential that the additional visual input would provide. A follow-up study could investigate into the possibility of adding speech-to-text or plain text input while keeping the complexity low.

1 Introduction

Conversational software is software that allows users to interact with a computer using either written or spoken natural language (Shawar and Atwell, 2003), usually to assist the user in accomplishing a task or acquiring information, in many cases only useful in one domain or area of expertise. There are two major approaches to conversational software, chatbots and dialogue systems.

1.1 Chatbots

The first of these two approaches, a chatbot or chatterbot (Mauldin, 1994), is a very simple system under the assumption that conversation is turn-based and that a next sentence is only dependent on the last thing the conversation partner said. These systems usually generate responses based by matching keywords or patterns in the user input with a pre-defined set of input/response pairs. (Shawar and Atwell, 2003) The success of these systems is entirely based on the size of their reply/response pairs and it fails as soon as any ambiguous input is encountered. For example, the sentence 'Can you explain this in more detail?' will never match the right response in all cases because it is impossible to determine what 'this' refers to from the sentence alone.

1.2 Dialogue Systems

The second approach to conversational software, dialogue systems, are rooted in a more rigorous scientific background, based on natural language processing and understanding. In these systems, input is parsed into a semantic representation and passed to a system called a dialogue manager which keeps track of the context of the conversation, basic facts and information such as who spoke last (Jönsson, 1995). With all this information a dialogue manager generates a response which can then be returned from the semantic representation to natural language. One such system is known as the information state approach (Traum and Larsson, 2003), which is a system that keeps track of the facts it knows and it is able to extract new information about the world and context from the conversation.

1.3 Problem statement

Both of these approaches to creating a conversational program have their advantages and disadvantages. The chatbot approach is easy to implement, but too simple for any 'real' conversation and it requires a lot of data in the form of reply/response pairs. The dialogue system approach is a lot more human-like in conversation, but requires a lot of specialized knowledge to create and is too complex to be feasible in most real-world systems. This aim of this project is to investigate the opportunity for enhancement of conversational chatbot dialogue, not by improving the dialogue system but by providing the chatbot with another input in addition to textual input. Such a system should

[1] University of Groningen, The Netherlands, email: m.g.meedendorp@student.rug.nl

have a performance more similar to that of a complex dialogue system but should be more versatile a simple chatbot system. It should be able to initiate a conversation and switch topics without user input.

We believe visual input would be the best suited for such a role because it requires only a camera and advancements in the field of visual processing have made simple object detection and classification relatively easy, allowing us to build on previous achievements which keeps the complexity low. A system with visual input would be able to start a conversation on its own about the things it's seeing without the user having to initiate the conversation, which is inevitable in systems which only use the conversation as input. It would also be able to switch topics by itself (e.g. saying 'Hi' when a person walks by).

2 Research Objectives

The main objective of this research is to build a conversation system with visual input, hopefully providing it with the simplicity of a chatbot system while improving the conversation towards that of a dialogue system.

The system's success will be judged by these research questions:

1. How can visual input be integrated into a chatbot system?
2. Does the system have any benefit to the quality of dialogue as measured by the following criteria?
   (a) Can the system start conversations on its own?
   (b) Can the system autonomously switch topics?
3. Is such a system still easier to build than a complex dialogue system?

3 Methods

Since the questions posed in section 2 are mainly open-ended and a simple yes/no answer would not suffice, a qualitative research design is appropriate.

From the research questions, there are two relevant aspects to the system. One is related to the user of the system, the other to the system developer. The user is relevant for research questions 2a and 2b, since they can be judged in context of the typical interaction between the user and the system. Research questions 1 and 3 are related to the system developer's perspective, because these questions can be answered in the context of the experience of the system developer in developing the system.

In order to answer the research questions in their relevant contexts, the two contexts will be individually assessed in the sections 4 and 5. In section 4, Implementation, the implementation of the system will be described, allowing us to answer research questions 1 and 3. In section 5, Testing, a typical user interaction with the system will be described, allowing us to answer questions 1 and 3.

4 Implementation

The system was implemented using ROS, the open-source Robot Operating System, which provides libraries for image capturing and processing, allows for distributed computing and various features helpful while debugging. It also handles separation of concerns by allowing different parts of a system to be split into so-called *packages* and allows communication between packages using *messages* with either a publish/subscribe or request/response mechanism. For more details see Quigley et al. (2009).

The system consists of three different computers: a Intel Euclid, an Ubuntu laptop with a Nvidia Geforce GTX GPU and a Sota humanoid robot with a Raspberry Pi 2B inside of it, combined with a separate keyboard for textual input and a webcam for visual input. All of these parts are set up in a shopping cart which allows the system to be moved around, a good way to test different visual environments. The Euclid is used as a ROS master and for person detection so it is mounted facing the person, while the Sota is facing sideways, looking out from the shopping cart at the environment. The webcam is facing the same direction as the Sota and the keyboard is located under the handlebar of the shopping cart. The laptop is placed on the bottom rack. Both the Euclid and the Sota are connected to the laptop, via a local Wi-Fi hotspot and an ethernet cable respectively. Both the laptop and the Euclid are running ROS, while the Sota is controlled over a TCP socket.

To build one distributed system on these different machines, the system was divided into four distinct modules, each of which was implemented in a separate ROS package. The main functions of these modules are visual input, speech output, dialogue management and Sota gaze direction. Each of these will be described in more detail. An overview the entire system can be found in Figure 1. The four packages are indicated as blue rectangles, with the name of the package in the center: ssd, commu_wrapper, dialogue and look_helper.

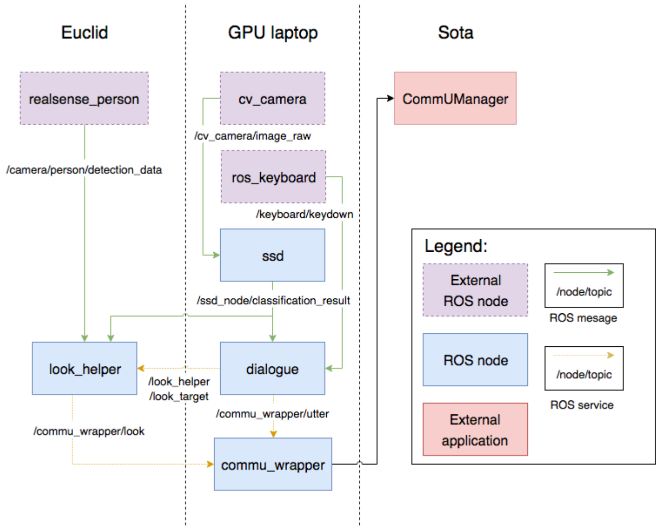

Figure 1. An overview of the system setup

The system's visual input is processed in the simplest package, ssd. It uses an object-detection and recognition neural network called Single-Shot Multibox Detector (abbreviated as SSD, hence the package name) as presented in Liu et al. (2016). The network used is pretrained on the PASCAL Visual Object Classes 2012 dataset, which contains 20 different classes of objects. It takes an image from the camera as input and outputs a list of detected objects and their bounding boxes. This output from the network is then passed along to other packages via a ROS topic. The ssd package uses input from a USB webcam, captured using the cv_camera package (Ogura, 2013), as can be seen in Figure 1.

The speech output of the system is handled in the commu_wrapper package. This package exposes CommUManager to the ROS system through two services. CommUManager is a process on the Sota providing text-to-speech and gaze control functionality.

The Sota's gaze direction is mainly determined in the look_helper package. This package translates between the Sota's coordinate system and the ROS tf system (Foote, 2013), allowing the Sota to look at locations published in the tf system, which the Euclid uses for person detections.

The dialogue package handles the dialogue management of the system. It is the central element of the system. It takes the visual input from the ssd package and keyboard input from ros-keyboard, and generates output in the form of requests for utterances at commu_wrapper and gaze positions at look_helper. Using the visual data, the package determines an appropriate pre-scripted dialogue based on the object category and priority. Priority is based on the size of the bounding box of the object to make the system more likely to talk about bigger objects. The system will choose the object with the highest priority to talk about, and it will try to interrupt the ongoing conversation if an object with a higher priority comes in.

Each of the 20 types objects from the dataset has an attached dialogue, as specified in a *dialogue library*. These libraries map a visual input label to a dialogue, either by using pre-scripted or dynamically generated dialogue. Dialogue libraries can easily be substituted to provide the system with new dialogues. Two different dialogues libraries were implemented in the system: one with a completely pre-scripted dialogue, one for all 20 classes of objects, and one that dynamically generates a small quiz-like game in which the robot asks the user whether he can find the objects that it is seeing, choosing from several different ways of phrasing and the questions and answers to generate a unique dialogue each time .

Dialogues are based on a set of actions. There are four defined actions. One action to pause the conversation, simply called sleep, one action to set a look target, named look, and two utter actions. The look action changes the gaze of the Sota in one of four possible directions: at the conversation partner, at the object that is being talked about at the moment, at a tf frame and at a random position, making the robot act like it's looking around. Of the two utter actions, one represents an utterance where a yes/no response is expected from the user and one for an utterance without any expected response. Each action provides the system with the next action to take, so each yes/no choice allows the dialogue to branch. Because an action provides a next action, the system is entirely adaptable to a more dynamic utterance generation approach, but for this proof-of-concept the dialogue is pre-scripted.

5 Testing

In this section, two typical interactions with the system will be described. These dialogues are from two different dialogue libraries, as described in section 4. Throughout the dialogue, notes will be given to explain what is going on in the system.

The first dialogue described was entirely scripted in advance, one short conversation for each possible object class, and provides insight into the conversation-starting and topic-switching mechanisms.

5.1 Pre-scripted dialogue

A cat walks in front of the webcam.
System: I see you have a cat.
I love cats!
I especially like black cats.
Do you like cats too?

The cat was classified by the SSD network and the relevant dialogue from the dialogue library was retrieved and used. Now, the system waits for a response from the user. The user responds by pressing the 'y' or the 'n' key on the keyboard.

User: *Presses the 'y' key on the keyboard.*
System: Good!

The system detects a chair

System: That's a nice chair.

At this point, a person walks by. Because people have a higher priority than other objects, the system will try to switch the conversation to one about people. Since the second sentence about the chair was already initiated, this will be uttered first.

System: I've always liked that kind of chair.

Because this sentence is an appropriate place to discontinue the chair dialogue (as indicated by a flag provided by the dialogue library), it will be stopped prematurely and the person dialogue will kick off.

System: Hey there, nice to meet you!
Can I ask you a question?

At this point in the conversation, a user response would be expected. The script is unable to handle dynamic responses, since it is impossible to pre-script a response for every possible user input, so the system simply waits for 2 seconds, allowing the user to answer.

System: Have you ever talked to a robot before?
User: *Presses 'n' on the keyboard.*
System: Well, there's a first time for everything.
Let's have a chat.

5.2 Dynamically generated dialogue

Initially, the robot is just looking around at the environment randomly. Then it detects a dining table.

System: *Looks at the conversation partner.*
Do you also see a dining table?
User: *Presses 'n' on the keyboard.*
System: *Looks at the dining table.*
I can help you. It's over here.
Looks around randomly after 3 seconds.

Here, the system is randomly choosing from a set of predetermined positive or negative replies. All other objects generate a similar dialogue, with the same question and one out of 8 responses.

6 Discussion

The objective of this research has been to show by implementation how a chatbot system can be equipped with visual input. This objective has been accomplished, answering research question 1 by providing a description of an implementation in section 4. With the details of the implementation of the system from section 4 and the description of the typical interaction with the system in section 5, the remaining research questions can also be answered.

6.1 System complexity

Research question 3 asks whether the system is still simpler than a sophisticated dialogue system. This is definitely the case, because it does not involve any natural language processing and understanding, while the added overhead of the visual input is relatively low. The visual input does, however, require a lot of processing power so it might not be practical in some cases. The main advantage of our system over an advanced dialogue system is that there is no linguistic knowledge involved in creating dialogue for the system, or the system itself.

6.2 Conversation quality

The last two research questions (2a and 2a) are easily answered using the results from section 5. There, the conversation is initiated by the system and a dialogue about a chair is interrupted by a new dialogue because a person was recognized. These results allow us to answer both research questions with a firm yes.

6.3 Further work

Improving chatbot systems' dialogue by adding visual input seems like a promising approach. Enabling the chatbot system to take initiative and start a conversation or switch a topic makes conversation feel a lot more natural.

The main improvement that could be made on the system in the future is in the area of dialogue generation, by replacing semi-scripted dialogue with a more sophisticated chatbot system. This is one of the main limiting factors of the system at the moment, since the dialogue is quickly exhausted. For a chatbot system to work in this configuration, though, the user must be able to give some kind of textual response, either by typing on a keyboard or via a speech-to-text function, thus allowing the user to provide a broader range of responses.

It might also be useful to train the object classification network on a different dataset with more distinct classes of objects or classes. This would allow the system to be adapted to different environments. An office environment might require recognizing computers and notebooks, for example, while books, tables and couches may be more relevant in a house setting.

7 Conclusion

The purpose of this project was to investigate the benefits of adding visual input to a chatbot system in order to improve dialogue capabilities by developing a prototype system. Even though the capabilities of the prototype are fairly limited, it shows that the presented approach is very promising. The system is easier to create than sophisticated dialogue systems and more capable in conversation than a simple chatbot.

REFERENCES

Foote, T. (2013). tf: The transform library. In *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, Open-Source Software workshop, pages 1–6.

Jönsson, A. (1995). A dialogue manager for natural language interfaces. *Proceedings of the Pacific Association for Computational Linguistics, Second conference*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). SSD: Single shot multibox detector. In *ECCV*.

Mauldin, M. L. (1994). Chatterbots, tinymuds, and the turing test - entering the loebner prize competition. *Proceedings of the National Conference on Artificial Intelligence*, (12/1):16.

Ogura, T. (2013). cv_camera - ROS Wiki. `https://wiki.ros.org/cv_camera`. [accessed 3 February 2018].

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.

Shawar, B. A. and Atwell, E. (2003). Using dialogue corpora to train a chatbot. In *Proceedings of the Corpus Linguistics 2003 conference*, pages 681–690.

Traum, D. R. and Larsson, S. (2003). The information state approach to dialogue management. In *Current and new directions in discourse and dialogue*, pages 325–353. Springer.