MULTIMODAL CONVERSATIONAL HELPBOT TO SUPPORT ROBOT ASSEMBLY TASK

Richard Tzong-Han Tsai[†] *Bo-Hao Chang*[†] *Cheng-Fu Yang*[‡] *Jeffrey Chiu*[§] *Hung-yi Lee*[‡]

[†]Dept. of Computer Science and Information Engineering, National Central University, Taiwan
[‡]Dept. of Electrical Engineering, National Taiwan University, Taiwan
[§]Dept. of Information Systems, Carnegie Mellon University, USA

ABSTRACT

Nowadays, question answering systems are becoming increasingly popular in the field of Natural Language Processing. In this research, we present the workflow to build a Multimodal Question Answering System that will help someone complete a task. We've demonstrated our work on the Meccanoid, a personal robot developed by Spin Master. When the user encounters problems in the assembly of the Meccanoid, they will ask our system, and our system will provide the best guide as the solution for their problems. In this paper, we propose a novel method that shows how to construct a question answering system similar to a FAQ in the task of the Meccanoid robot assembly, including the stages of data collection, user intent definition, and classification. Furthermore, we introduce a multimodal architecture for solving the bottleneck which the traditional single-modality system may encounter. The experimental results show that the combination of visual and textual context enhances the performance of intent classification work by 18%. The workflow presented should be able to generalize to other domains depending on the requester's demands, hopefully adapting to the smart manufacturing field.

1. INTRODUCTION

While researchers have proposed many deep learning models for building dialogue systems, most of these models are only focused on how to beat the state-of-the-art technologies that are currently used for the public datasets and NLP tasks. Not many of these papers address the difficulty of adaptation while trying to account for large-scale training data, and that is an issue as these predictive models are only as good as their training with robust data allows them to be. We, however, have addressed this issue and will be introducing the way in which we collected data through a crowdsourcing platform: the Amazon Mechanical Turk (MTurk).

Generally, FAQs are set up with prior experience running the operation, and the content of an FAQ is normally written in formal writing, not in normal prose. Thus, we wanted to collect various ways in which our base FAQ problems could be asked or presented. To do this, we placed all questions from our FAQ into individual categories and assigned initial user questions into such categories. After assigning initial user questions, we would start training a classifier to 'classify user questions into the right categories.'

Object detection is another popular topic, and many papers have shown promising result. Some models have shown good performance on real time object detection [1, 2] when trained on large datasets, such as the Pascal VOC dataset [3], but they fail to reproduce the same result on small object. Others proposed improved method [4, 5, 6] which used similar concept to feature pyramid networks [7] to generate better performance in small object detection. In our task, considering that our task needs to immediately respond to our user once the model has fetched user questions, real time object detection is desired. Therefore, we would prefer methods like SSD or YOLO [8, 1].

In this paper, we will present our task called the Meccanoid Robot Assembly to showcase our methods for building a lightweight task-oriented dialogue system that can generalize to other instruction-giving tasks. The Meccanoid is shown in Figure1. Since the Meccanoid has plenty of components, to reduce the complexity to a reasonable level in the pilot study, we make the users assemble the Meccanoid themselves without an instruction book but with sub-assembly of its main components, that are all finished. However, during the final assembly task, users would still encounter many problems such as where to insert screws, and whether they were being put in the correct hole or not. This is where our machine testing came in. When the user has problems, they will speak about the problem to the machine, and the machine will have a chat with the user until the machine gets a description of what is needed to provide the best known solution to the problem. For more concrete illustrations, please check our demo here: http://youtu.be/M55at1PkY14.

Besides doing knowledge transfer and building a QA system itself, another point we wanted to focus on in this paper is the issue of multimodality [9, 10] . We know that traditional systems can only process the information of single modalities at a time, thus restricting its abilities to cope with more complex environments. Hence, in our robot assembly task, an user being unable to describe his or her situation in a clear way could prove troublesome, as it is difficult for our

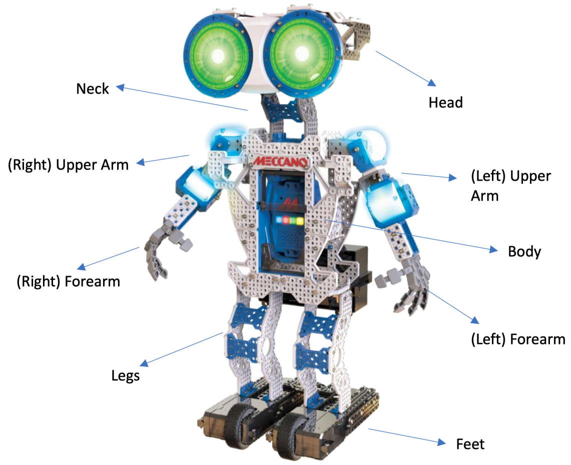


Fig. 1. The nine main body parts of the Meccanoid robot

text-only model to do all classification work. Thus, we have come up with a method to incorporate other modalities to support our system. With the help of the visual context, and we know that visual features combined with language modeling have shown good performance in several task like image captioning [11, 12, 13, 14] and in question answering on images [15, 16, 17]. Therefore, we hope to parse through user questions more precisely. Our Visual & Textual Sensitivity model can classify user questions considering the result of object recognition. Compared with the text-only model, this has enhanced performance by around 15%.

2. DATA COLLECTION

In this study, we need to construct a dialogue system to help the novice to achieve a task: final Meccanoid robot assembly. This task is to ask the novice to assemble the main components of the Meccanoid robot, including head, neck, body, legs, feet, and hands. In this section, we describe how we collected our intent/question data in two steps: a pilot study and user question collection.

2.1. Pre-Data Collection: Wizard-of-Oz Pilot Study

We start by carrying out a pilot study to investigate what kinds of problems a user may encounter in our final assembly task and to compile a core set of question intents. The study is based on a Wizard-of-Oz experiment that simulates our machine help agent while subjects carry out the robot assembly task. For subjects, we recruit people without any experience in assembling robots. However, as our agent does not yet exist at this stage, we have a team member with expertise in robot assembly help the subject with the task. To simulate human-machine interaction, our team member remains in a different room and can only communicate with the subject via Skype voice chat. At the start of the experiment, the assistant introduces our final assembly task and encourages the subject to ask questions if they encounter any problems. During the assembly process, the human assistant remains on call to answer any questions that come up.

For the experiment setup, we provide a laptop with which the subject can communicate with the agent, and we film the assembly area from two perspectives in full-HD/30fps. In order to ensure high-quality voice data, we set up an extra 2-channel/48kHz microphone to record the subject.

On average, each subject took 40 to 50 minutes to finish the task. We collected data from 15 subjects in total. Two QA examples between the assistant and subject are listed as follows.

- Example 1:
  - Q : Which direction do the screws have to go in? Doesn't matter?
  - A : (Show picture) You can check this and direction is from top to bottom.
- Example 2:
  - Q : There are two holes I can plug screw S2 and is it ok to put either side of them?
  - A :(Show picture) You only have to pick one hole to put S2 in.

After the pilot study experiments are complete, we must sort through the data. Watching the assembly videos, we transcribe all user questions and determine which overlap and which are unique. According to our observations, user questions can be clustered into 21 scenarios. For each scenario, we pick the most representative question and then add it to our core list of question intents.

2.2. User Question Collection

To train our conversational model, we must collect data. However, the cost of collecting directly from the workbench is too high. Therefore, we designed a Human Intelligence Task (HIT) to collect other alternative questions for each core question intent on Amazon Mechanical Turk (MTurk).

We created our MTurk HIT for the Meccanoid robot assembly task using HTML and JavaScript with our own template. Based on the pilot study results, we have 21 core question intents. The goal of our HIT is to generate 100 different oral variations of each question, for a total of 2100 questions.

We first introduce the final assembly task in a short paragraph. For each question intent scenario, we provide the turker a video that demonstrates the scenario in which the assembly problem was presented and provide sample questions collected during the pilot study for reference. The turker must submit a new variation for each core question intent (scenario), for a total of 21 sentences per HIT.

Table 1. Question Intent Dataset Statistics

Attributes	Statistics
# of Turkers	180
# of scenarios	21
# of collected questions	3769
# Average work time per worker(min)	17

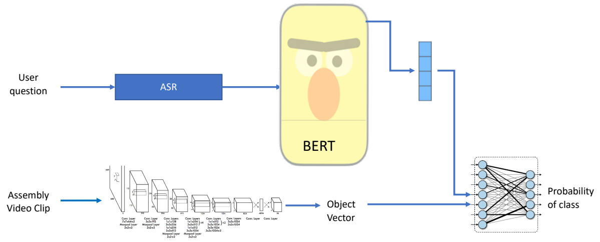


Fig. 2. Our Multimodal Architecture

To prevent turkers from copying the example questions directly, they must click the check button before submitting. If the similarity of a submitted question to any of the examples or previously submitted questions exceeds 80%, the submitted question is rejected. We use cosine similarity because it is easy to implement and fast. All 21 questions must be filled in with original variations, or the HIT cannot be submitted. For each 21-sentence HIT, we pay the worker \$3.75.

After removing incorrect sentences, we collected a total of 3,769 user questions. Table 1 summarizes the basic statistics of our question intent dataset.

3. MODEL

Figure 2 shows our multimodal architecture, we call it the Visual & Textual Sensitive Intent Classifier. When the user asks a question, the question utterance is converted to text by the Google automatic speech recognition (ASR) service and then transformed in to a representation vector by BERT [18], which is one of the state-of-the-art language model.

At the same time, the video clip is processed by our visual state discriminative model (V-SDM) to generate an object vector indicating which parts are currently held by the user.

Then, the sentence representation vector and the object vector are concatenated to form a vector. This vector is then fed into an MLP classifier to determine its intent. In the following section, we will introduce the details of our V-SDM model.

3.1. Visual State Discriminative Model

The visual stage discriminative model (V-SDM) is an end-to-end model that analyzes a video clip of the robot assembly process and outputs an object vector indicating which components were detected.

For each input frame, the model generates bounding boxes at three different scales by the Yolo v3 network [19], which is a deep convolutional neural network. Each box is assigned with probabilities of all possible nine objects. Therefore, the object with the highest probability is the predicted object for that box and we denote the probability as the box's max probability. Boxes with max probability lower than a threshold will be discarded. The probability vectors of all selected boxes are treated as the input of the last softmax layer to generate the probability vector for this frame.

Then, the model accumulates the probability vectors of all the frames in this clip as the input to generate the video's probability vector. For each dimension of the probability vector, if the value is above the threshold, then the same dimension of the object vector is set to 1, otherwise, the value is set to 0.

3.2. Visual Training Data Collection

We use the Workbench to collect visual data. For the training data of object recognition model: we shoot pictures of every angles of all robot components. This is a widely adopted technique called data augmentation. For each object o, when moving o, the others are fixed. 8 videos in different directions are recorded. In total, 72 video clips were recorded, each contains 200 to 250 frames.

4. EXPERIMENTS

4.1. Test Set Compilation

For each of the 21 question intent cores, we have video recordings of the user asking the question. We then select 798 question variations as the test set. For each question in the test set, we randomly select 200 250 frames from the video of its core question intent as the question's corresponding video. Table 2 shows three data instances from our test set.

4.2. Evaluation Metrics

In addition to accuracy, we also use mean reciprocal rank (MRR) as another metric. Reciprocal rank (RR) is used to evaluate the system's ability to output a list of instances given a query. MRR is the extension of RR for a set of queries. It is defined as:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \qquad (1)$$

where Q is the set of queries; i is a query, $rank_i$ is the first correct answer's rank for query i. This metric is useful for gaining a general understanding of how accurately the system ranks the correct answer.

Table 2. Example data instances in our test set

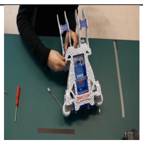
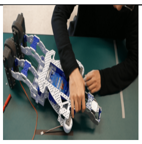
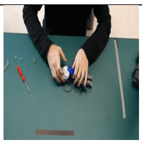
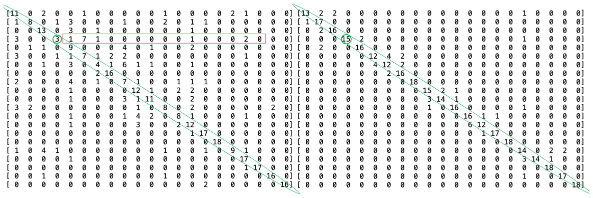
user question	Which way do I put the screws for locking it?	Dose it matter which direction the neck faces?	Does it matter which way I snap the arm into the lock?
formal question	Is there a right direction to lock the screws?	Does the neck have front side or back side?	Should I embed it?
class	1	8	12
video fraction			

Table 3. Performance of text-only and text+visual intent classification

	Accuracy	MRR
Baseline MLP	0.61	0.74
MLP+BERT	0.70	0.79
Visual+MLP(W2V)	0.85	0.91
Visual+MLP(BERT)	0.88	0.93



```
[11  0  2  0  0  1  0  0  0  0  0  1  0  0  0  0  2  1  0  0  0] [13  2  2  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0]
[ 1  8  0  1  3  0  0  0  1  0  0  2  0  1  1  0  0  0  0  0  0] [ 1 17  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0 13  0  3  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0] [ 0  2 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 3  0  0  3  1  7  1  0  0  0  0  0  0  1  0  0  0  2  0  0  0] [ 0  0  0 15  2  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0]
[ 0  1  1  0  9  0  0  0  4  0  1  0  0  2  0  0  0  0  0  0  0] [ 0  2  0  0 16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 3  0  0  1  1  7  1  2  2  0  0  0  0  0  0  0  0  1  0  0  0] [ 0  0  0  0  0 12  4  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  1  0  3  0  4  1  6  1  1  0  0  1  0  0  0  0  0  0  0] [ 0  0  0  0  0  4 12  2  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  2 16  0  0  0  0  0  0  0  0  0  0  0  0  0] [ 0  0  0  0  0  0  2 16  0  0  0  0  0  0  0  0  0  0  0  0  0]
[ 2  0  0  0  4  0  1  0  7  1  0  0  1  1  1  0  0  0  0  0  0] [ 0  0  0  0  0  0  0  0 18  0  0  0  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  1  0  0  0  0 12  1  0  2  2  0  0  0  0  0  0  0] [ 0  0  0  0  0  0  0  0  0 15  2  1  0  0  0  0  0  0  0  0  0]
[ 0  0  0  0  1  0  0  0  3  1 11  0  0  2  0  0  0  0  0  0  0] [ 0  0  0  0  0  0  0  0  0  3 14  1  0  0  0  0  0  0  0  0  0]
[ 3  2  0  0  0  0  0  0  0  1  0  8  0  2  0  0  0  0  0  2  0] [ 0  0  0  0  0  0  0  0  0  1  0 16  0  0  0  0  1  0  0  0  0]
[ 0  0  0  0  1  0  0  0  1  4  2  0  8  1  0  0  0  1  0  0  0] [ 0  0  0  0  0  0  0  0  0  0  0  0 16  1  1  0  0  0  0  0  0]
[ 0  0  0  0  1  0  0  0  0  3  0  0  2 12  0  0  0  0  0  0  0] [ 0  0  0  0  0  0  0  0  0  0  0  0  6 12  0  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  1 17  0  0  0  0  0  0] [ 0  0  0  0  0  0  0  0  0  0  0  0  0  1 17  0  0  0  0  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 18  0  0  0  0  0] [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 18  0  0  0  0  0]
[ 1  0  4  1  0  0  0  0  0  0  0  1  0  0  1  0  9  1  0  0  0] [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 14  0  2  2  0]
[ 0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0 17  0  0  0] [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  3 14  1  0  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 17  0  0] [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 18  0  0]
[ 0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0 16  0] [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0 17  0]
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0 16] [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 18]
```

Fig. 3. Analysis: Confusion Matrix

4.3. Results

We compare text-only and text+visual classifiers. For the baseline text-only system, we use the MLP classifier with a pre-trained Word2Vec model. Then, we replace Word2Vec with BERT, which dramatically increases accuracy and MRR by 9% and 0.05, respectively.

To create the text+visual system, we concatenate the object vector generated by the V-SDM model with the embedding vector generated based on text. Comparing the results, we find that the visual information boosts the performance of the text-only models. For the MLP+Word2Vec model, it increases the accuracy and MRR by 24% and 0.17, respectively. For the MLP+BERT model, it achieves an accuracy of 88% and an MRR of 0.93, outperforming the text-only baseline by 27% and 0.19 in terms of accuracy and MRR.

4.4. Error Analysis and Discussion

The classification errors are listed in the confusion matrix (Figure 3) below. Comparing the text-only with visual+text classifiers, we can see that the visual+text classifier greatly improves classification performance. Take intent 4 (circled in green) for example. The assembly intent of intent 4 is screw direction. In the text-only confusion matrix, intent 4 is incorrectly classified 7 times as intent 6, whose assembly intent is also screw direction. This verify the assumption we proposed in previous section again: The scenarios sharing the same assembly intent make classifier fail to discriminate them if only through textual context. While we incorporate the visual context as another feature for classification, the situation becomes obvious. We can check the example of class 4 and class 6 again. In fact, the user is assembling feet & body in class 4 and neck & body in class 6. The information of object vector (visual context) indicates the user state clearly. Even if the user question is ambiguous, our V+T model can still make the right decision.

5. CONCLUSION

In this paper, our contribution has three folds. First, we demonstrate how to develop a helper bot from the scratch. We first conducted a pilot study based on the Wizard-of-Oz approach to collect FAQs. These FAQs were then clustered into core intents. Then, based on these FAQs, we design human intelligence tasks on the Amazon Mechanical Turk platform to create more variants for each core intent. These data are used to generate textual-based features. Secondly, we design a multimodal intent classifier combining the text and visual modalities. Finally, we design procedures to collect visual training data and the multimodal test data. Experimental results show that our V+T modal outperforms the state-of-the-art Text-only model by 18% and 0.14 in terms of accuracy and MRR, respectively. From the detailed analysis we found that the V+T model really mitigates the errors of the text-only model.

6. REFERENCES

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[4] Ping Hu, Bing Shuai, Jun Liu, and Gang Wang, "Deep level sets for salient object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2300–2309.

[5] Jianan Li, Xiaodan Liang, Jianshu Li, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan, "Multistage object detection with group recursive learning," *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1645–1655, 2017.

[6] Yun Ren, Changren Zhu, and Shunping Xiao, "Small object detection in optical remote sensing images via modified faster r-cnn," *Applied Sciences*, vol. 8, no. 5, pp. 813, 2018.

[7] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016, pp. 21–37.

[9] Gunther Kress and Theo Van Leeuwen, *Multimodal discourse: The modes and media of contemporary communication*, Edward Arnold, 2001.

[10] Carey Jewitt, *The Routledge handbook of multimodal analysis*, Routledge London, 2009.

[11] Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell, "Language models for image captioning: The quirks and what works," in *arXiv preprint arXiv:1505.01809*, 2015.

[12] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[13] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al., "From captions to visual concepts and back," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1473–1482.

[14] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*, 2014, pp. 647–655.

[15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.

[16] Arijit Ray, Gordon Christie, Mohit Bansal, Dhruv Batra, and Devi Parikh, "Question relevance in vqa: identifying non-visual and false-premise questions," *arXiv preprint arXiv:1606.06622*, 2016.

[17] Mateusz Malinowski and Mario Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," in *Advances in neural information processing systems*, 2014, pp. 1682–1690.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.