mForms : Multimodal Form-Filling with Question Answering

Larry Heck, Simon Heck, Anirudh Sundar

Georgia Institute of Technology

{larryheck, sheck6, asundar34}@gatech.edu

Abstract

This paper presents a new approach to form-filling by reformulating the task as *multimodal natural language Question Answering (QA)*. The reformulation is achieved by first translating the elements on the GUI form (text fields, buttons, icons, etc.) to natural language questions, where these questions capture the element's multimodal semantics. After a match is determined between the form element (Question) and the user utterance (Answer), the form element is filled through a pre-trained extractive QA system. By leveraging pre-trained QA models and not requiring form-specific training, this approach to form-filling is zero-shot. The paper also presents an approach to further refine the form-filling by using multi-task training to incorporate a potentially large number of successive tasks. Finally, the paper introduces a multimodal natural language form-filling dataset Multimodal Forms (mForms), as well as a multimodal extension of the popular ATIS dataset to support future research and experimentation. Results show the new approach not only maintains robust accuracy for sparse training conditions but achieves state-of-the-art F1 of 0.97 on ATIS with approximately 1/10th the training data.

Keywords: Form-Filling, Question Answering, Multimodal Machine Learning

1. Introduction

The last decade has seen the development and broad deployment of digital assistants (DAs) including Siri, Cortana, Alexa, Google Assistant, and Bixby. A primary component of DAs is Natural Language Understanding (NLU) - understanding the meaning of the user's utterance. Referring to Figure 1, the NLU task determines the domain of the user's request (e.g., travel), the user's intent (e.g., find_flight) and information-bearing parameters commonly referred to as semantic slots (e.g., City-departure, City-arrival, and Date). The task of determining the semantic slots is called slot filling (Tur and De Mori, 2011). In this paper, we address a related but distinct task - *form-filling*, where the DA processes the user requests to act on form elements (fill text fields, click buttons and icons, etc.) on Mobile Apps or web pages. Equipping DAs with the ability to simultaneously parse visual semantic information and contextual dialogue enhances their ability to understand and act on information across multiple modalities. This type of multimodal interaction through conversations is currently an open problem and an active area of research (Sundar and Heck, 2022).

Early methods for the related area of semantic slot filling used recurrent neural networks (RNNs) (Mesnil et al., 2014), then progressed to long short-term memory (LSTM) neural networks (Liu and Lane, 2016), and more recently transformer-based approaches (Chen et al., 2019).

Dynamic deep learning approaches, while achieving high slot-filling accuracy, demand extensive domain-specific supervised training data. This poses challenges for applications with limited access to extensive data, such as AI skill develop-
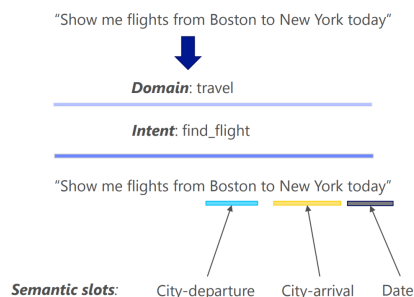

Figure 1: An example semantic representation with domain, intent, and semantic slot annotations.

ment for DAs, limiting the broad expansion of AI skills to adequately cover the long tail of user goals and intents.

Prior work has focused on developing models and approaches that require less supervised training data. Zero and few-shot learning methods have been developed across NLP tasks (Dauphin et al., 2013; Yann et al., 2014; Upadhyay et al., 2018). Methods can be broadly categorized into transfer learning (Jaech et al., 2016; El-Kahky et al., 2014; Hakkani-Tür et al., 2016), sequential learning (Bapna et al., 2017a), reinforcement learning (Liu et al., 2017; Kumar et al., 2017; Shah et al., 2016) and synthetic training (Xu et al., 2020; Campagna et al., 2020).

In many cases, the user interacts with an App screen or web page and, therefore, uses multiple modalities such as voice, vision, and/or touch (Heck et al., 2013; Hakkani-Tür et al., 2014; Li et al., 2019; Selvaraju et al., 2019; Zhang et al., 2020; Xu et al., 2021; Reichman et al., 2023; Zhang et al., 2019; Sundar and Heck, 2023; Reichman and Heck, 2023; Sundar et al., 2024). For

11262

these settings, zero- and few-shot learning can be achieved by leveraging the semantics contained in the screen. In (Bapna et al., 2017b), the authors incorporated visual slot names or descriptions in a domain-agnostic slot tagging model called a Concept Tagger. The Concept Tagger models the visual slot description (e.g. "destination") as a Bag-of-Words (BOW) embedding vector and injects a Feed-Forward network inside the original deep LSTM network to process the user's utterance (e.g., "Get a cab to 1945 Charleston"). Results showed the inclusion of slot descriptions significantly outperformed the previous state-of-the-art multi-task transfer learning approach (Hakkani-Tür et al., 2016).

The Concept Tagger (Bapna et al., 2017b) is limited in several ways. First, the BOW semantic representation of the visual slot description is static and does not model the dynamics of the description language. Second, the method is limited to only visual slots with text descriptions and does not incorporate other semantic information from the visual elements (i.e., is the element a form field or a radio button with choices). Third, the Concept Tagger incorporates multi-task learning only through the visual slot description.

This paper[1] addresses all three limitations of the Concept Tagger. To address these limitations, the next section describes a new approach that formulates multimodal form-filling as Question Answering (QA). This approach also extends more recent work on text-based slot filling as QA (Levy et al., 2017; Du et al., 2021; Fuisz et al., 2022) by developing a much broader, multimodal computer vision-based approach. The extension to a multimodal approach is required in form-filling where the QA formulation must cover all of the 25 UI component categories, 197 text button concepts, and 99 icon classes.

In the Experiments Section, we introduce a new corpus collected for multimodal form-filling called the Multimodal Forms (mForms) dataset as well as an extension of the ATIS (Tur et al., 2010) dataset as a simulated form-filling task. We compare the new zero-shot multimodal form-filling QA approach to competing methods on this new corpora. Finally, we summarize our findings and suggest the next steps in the Conclusions and Future Work Section.

2. Approach

2.1. Multimodal form-filling

The foundation of the approach presented in this paper is the utilization of deeper semantics *in the visual representation of the form* on the user's screen. While previous form-filling methods
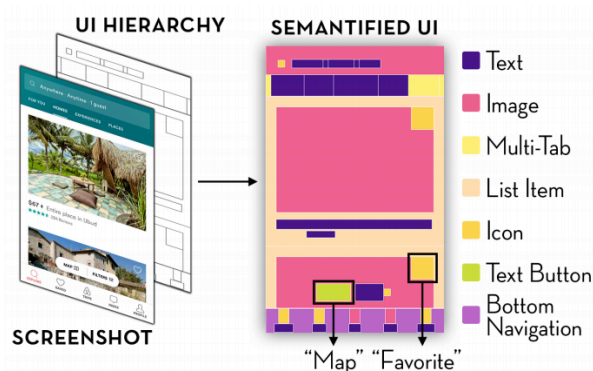
Figure 2: Semantically annotated mobile Graphical User Interface (GUI) using computer vision to identify 25 UI component categories, 197 text button concepts, and 99 icon classes (Liu et al. 2018)

treated the form label as a classification tag with no semantic information, the approach of this paper extracts meaning from the visual slot representation. By formulating the form field description as a Question and the user's utterance as the Paragraph, we can directly utilize transformer-based extractive Question Answering (QA) models (Lan et al., 2019). The Start/End Span of the extracted Answer is used to fill the appropriate content in the web form. We call our approach *Multimodal form-filling as Question Answering (QA)*, which we will henceforth refer to by *mForms as QA*.

In addition to the lexical semantics contained in the text field description, the type of the visual graphical user interface (GUI) element on the App or web page provides additional semantic information. The set of GUI design elements of a mobile App that are available to translate into questions are shown in Figure 2. In our approach, the GUI design elements are automatically classified via a convolutional deep neural network computer vision system trained on the RICO dataset as shown in Figure 9 of the Appendix (Deka et al., 2017; Liu et al., 2018). The computer vision classifier identifies 25 UI component categories (e.g., Ads, Checkboxes, On/Off switches, Radio Buttons), 197 text button concepts (e.g., login, back, add/delete/save/continue), and 99 icon classes (e.g., forward/backward, dots, plus symbol). Our implementation as described in (Liu et al., 2018) has a 94% classification accuracy.

In mForms, rules are used to translate each GUI design element into an appropriate question. Each type of GUI design element has a unique rule type that triggers depending on its visual presence on the GUI. Figure 3 shows an example of these GUI elements, their associated rule templates, and example questions and user utterances. If multiple GUI design elements are visible, then multiple translation rules fire, generating simultaneous questions to be paired with the user's utterance.
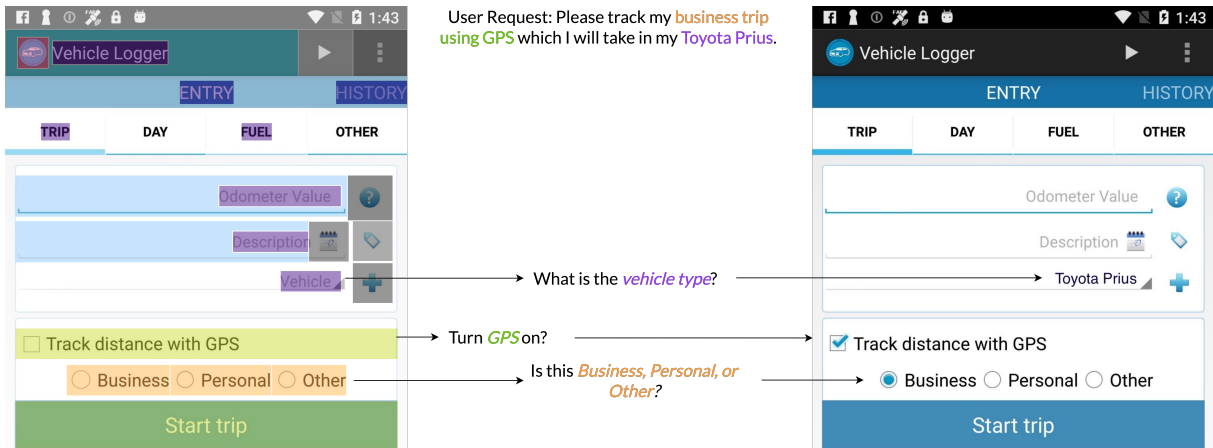
Figure 3: The mForms pipeline. The rule template uses the semantified UI to trigger a question template. The visual information of the GUI element drives the generation of the actual question. Then, using the user's request as evidence, the questions are answered to fill the form with the appropriate information.

For example, GUI elements that are classified as simple text fields trigger a rule that generates a question template "What is the *Text_Field*?". Figure 3 shows simple text fields in the Michaelsoft Vehicle Logger App. The first text field is "Vehicle". In this case, the rule recognizes command and generates a question "What is the vehicle type?". Given a user utterance "Please track my business trip using GPS which I will take in my Toyota Prius.", the Question-Answering system extracts the *answer* to this question as "Toyota Prius".
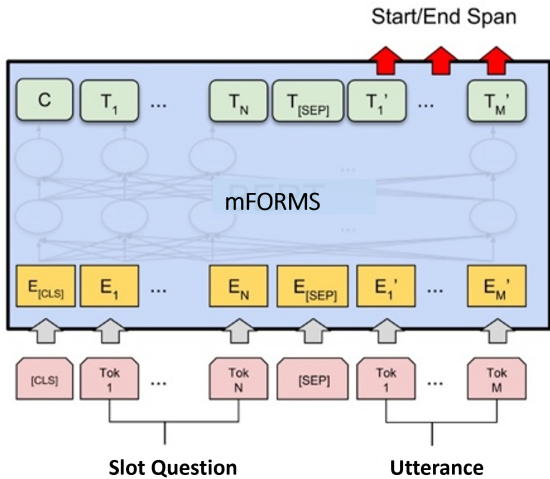
2.2. Single- and Multi-Task Training

Our mForms as QA method is shown in Figure 4. It can be formulated as both single- and multi-task training. The Single-task (ST) model is initialized as a general-purpose QA trained with SQuAD2 (Rajpurkar et al., 2018). Used as-is, this model is *zero-shot* for form-filling. The model can be fine-tuned with supervised (annotated) form-filling data from the visual App or web page GUI.

In contrast to Single-task training, Multi-task (MT) training incorporates form-filling training sets across multiple tasks with each training further refining the model. Similar tasks represented by common domains can be grouped for successive fine-tuning stages. For example, flight reservation form-filling Apps could be successively refined using the first $N-1$ Apps with the Nth App used as the final fine-tuning stage. The potential advantage of the MT approach is the required amount of annotated supervised training data becomes less with each new task refinement stage.

Figure 4: mForms as QA Approach

3. Experiments

3.1. Setup

Our base QA system is based on the Pytorch implementation of ALBERT (Lan et al., 2019) [2]. We use the pre-trained LM weights in the encoder module trained with all the official hyper-parameters[3].

3.2. Multimodal Forms Dataset

Amazon Mechanical Turk (AMT) was used to collect Multimodal Forms (mForms) - a dataset to support multimodal form-filling research[4]. The AMT crowd workers were asked to formulate requests to mobile App screens from three Apps in

---

the RICO dataset: Vehicle Logger from Michael-soft, United Airlines flight search, and Trip Advisor. The UIs of each App with GUI elements semantically annotated by the computer vision system described earlier are available online.[5] More details on the mForms dataset are in given in the Appendix.

3.3. Simulated ATIS Form-Filling Dataset

The ATIS dataset is a widely used NLU benchmark for users interacting through natural language with a flight booking system (Tur et al., 2010). To use ATIS for mForms as QA, we extended the dataset in several ways. First, as shown in Figure 5, each slot is treated as a simulated visual Text Field where the information of the slot tag is displayed in an App with a simple form. As is the case with Text Fields, each slot was reformulated as a natural language Question. For example, the ATIS slot tag "B-aircraft_code" is translated into the question "What is the aircraft code?". This modified dataset will be called "ATIS form-filling".

```
B-aircraft_code              what is the aircraft code?
B-airline_code               what is the airline code?
B-airline_name               what is the airline name?
B-airport_code               what is the airport code?
B-airport_name               what is the airport name?
B-arrive_date.date_relative  what is the relative arrival date?
B-arrive_date.day_name       what is the day name of the arrival date?
B-arrive_date.day_number     what is the day number of the arrival date?
B-arrive_date.month_name     what is the month name of the arrival date?
B-arrive_date.today_relative what is the arrival date relative to today?
B-arrive_time.end_time       what is the end time of the arrival?
B-arrive_time.period_mod     what is the time period modification of the arrival?
B-arrive_time.period_of_day  what is the period of day of the arrival time?
B-arrive_time.start_time     what is the start time of the arrival?
B-arrive_time.time           what is the arrival time?
B-arrive_time.time_relative  what is the relative arrival time?
B-booking_class              what is the booking class?
B-city_name                  what is the city name?
B-class_type                 what is the class type?
B-compartment                what is the compartment
```

Figure 5: Simulated ATIS form-filling fields translated to Questions

Table 1 summarizes the three Visual App datasets as well as ATIS form-filling with example utterances from each dataset. Table 6 in the Appendix shows the types of slots annotated in each dataset. ATIS form-filling has the largest number of slot types at 83.

Table 2 summarizes F1 scores (harmonic mean of precision and recall) on the 3 new mForms datasets and the ATIS form-filling dataset. For comparison, the F1 score is given for the joint slot and intent model (JB) given in (Chen et al., 2019). The new mForms as QA approach presented in this paper consistently outperforms the JB slot filler. While the JB slot filler requires at least 100 training samples on the Vehicle Logger App, the mForms as QA approach maintains the F1 score even for only 0 and 5 training samples. These results suggest the semantic information contained in the mForms is particularly important for sparse training conditions.

With more training data, is interesting to note that the accuracy of the new mForms as QA approach also achieves one of the best published F1 measures at 0.97 on the ATIS dataset. For comparison, the mForms as QA system was only trained on 500 samples for this case as compared to the full training set of ATIS at over 4400 samples. This suggests that the injection of simulated mForm and the subsequent generation of questions for the QA system is effective at reducing the amount of training data required to yield high accuracy. This characteristic of mForms as QA makes the approach especially attractive for commercial digital assistants given the industry's reliance on third-party developers who are often not highly skilled in NLU.

To examine the effect of visual semantics in a more controlled experiment, questions generated in the new mForms as QA approach were replaced with tag symbols, where the tag symbol had no semantic information (e.g., "XYZ"). Otherwise, "No Visuals" is the same model as mForms as QA. Results for the Vehicle Logger App are shown in Table 3 comparing the No Visuals approach to two conditions from the mForms as QA approach (1) Text Only - all visuals are treated as simple Text Fields and other GUI elements are ignored (2) All GUI elements are used. The training samples were randomly chosen across all 10 slot types from the complete set of 500 utterances. Larger differences are observed in sparse training conditions where the No Visuals approach largely falters.

Given the mForms as QA approach incorporates multi-task (MT) training, an interesting question to answer is whether the MT training transfers knowledge *across domains*. Table 4 shows results for the cross-domain case: fine-tuning on the ATIS form-filling dataset followed by another iteration of fine-tuning on data from the Vehicle Logger App. The effect of MT is more pronounced in the sparse training cases with an improvement from 0.48 F1 to 0.52 F1 at zero-shot training and an improvement of 0.46 F1 to 0.60 F1 with 5 training samples. These results suggests cross-domain concept learning is occurring for these training conditions.

Finally, Table 5 shows zero-shot F1 scores on the mForms datasets for our new approach when varying the number of visual GUI elements that are displayed to the user. For example, when 2 visual elements are displayed, the model must not only parse the slots from the utterance for one of the visual elements but also correctly reject the filling of slots into the other element. For the mForms datasets, the models degrade gracefully. This robustness is likely the result of the initial fine-tuning on the SQuAD2 dataset which is trained to reject

---

Visual App	Sample Utterance	# Utterances
Vehicle Logger	"Please activate GPS tracking and log my car trip"	850
United	"Book a Flight from California to arizona on august 15th 2020"	850
ATIS form-filling	"I live in Denver and I'd like to make a trip to Pittsburgh"	4478
Trip Advisor	"Please book a 5 star hotel in Atlanta Georgia"	803

Table 1: Sample utterances from each domain

# train samples	0		5		50		100		500	
Domain	JB	mForms	JB	mForms	JB	mForms	JB	mForms	JB	mForms
Vehicle Logger	0.00	0.48	0.00	0.46	0.48	0.73	0.45	0.80	0.78	0.87
ATIS form-filling	0.00	0.60	0.00	0.74	0.66	0.88	0.77	0.93	0.91	0.97
United	0.00	0.40	0.00	0.44	0.37	0.58	0.44	0.72	0.51	0.74
Trip Advisor	0.00	0.52	0.00	0.47	0.18	0.63	0.53	0.66	0.59	0.66

Table 2: Weighted token F1 (harmonic mean of precision and recall) scores. The table shows the baseline (JB) as detailed in (Chen et al., 2019) versus our new mForms as QA approach.

# train samples	0	50	100	500
No Visuals	0.01	0.29	0.32	0.71
Text Visuals	0.36	0.69	0.71	0.88
GUI (all) Visuals	0.48	0.73	0.80	0.87

Table 3: F1 results showing effects of visual semantics on the Vehicle Logger App. The row labeled Text Visuals shows the results of our mForms as QA method with every visual element treated as a simple text field. GUI (all) Visuals leverage the full semantic information contained in the visual GUI elements for mForms as QA.

# elements	1	2	3	4	5
Vehicle Logger	0.52	0.51	0.49	0.49	0.46
ATIS form-filling	0.60	0.58	0.56	0.53	0.52

Table 5: Zero-Shot Slot F1 scores on the Vehicle Logger and ATIS form-filling datasets for varying numbers of visual elements shown to the user simultaneously

# train samples	0	5	100	500
Vehicle Logger	0.48	0.46	0.80	0.87
+ATIS form-filling	0.52	0.60	0.80	0.89

Table 4: Results for multi-task training (MT) across multiple domains. The first row shows F1 scores for the Vehicle Logger dataset for various amounts of training data. The second row shows the effect of fine-training the SQuAD2 model with ATIS form-filling before training with the Vehicle Logger data.

false questions - questions that do not have a correct answer to extract from the given Paragraph.

4. Conclusions and Future Work

This paper presented a new approach to filling GUI forms by reformulating the problem as a multimodal natural language Question Answering (QA) task. The reformulation is achieved by first translating the elements on the GUI form (text fields, buttons, icons, etc.) to natural language questions, where these questions capture the element's multimodal semantics. These questions are paired with the user's utterance and answers are extracted from the utterance using a Transformer-based Question-Answering system. An approach to further refine the model is presented that facilitates transfer learning across a large number of tasks. The paper also presented mForms - a multimodal form-filling dataset to support future research.

Results show the new approach not only maintains robust accuracy for sparse training conditions but achieves state-of-the-art F1 of 0.97 on ATIS with approximately 1/10th of the training data.

Future work will extend mForms as QA to a broader set of visual GUI screens across both mobile Apps and web pages. In addition, we plan to explore improved rejection methods for screens with high-density competing visual GUI elements. Lastly, while mForms as QA uses a BERT-based architecture for comparison with prior work, future work will explore ways to leverage generative models such as GPT3.5-4/T5/BART.

5. Bibliographical References

Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2017a. Sequential dialogue context modeling for spoken language understanding. *arXiv preprint arXiv:1705.03455*.

Ankur Bapna, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2017b. Towards zero-shot frame semantic parsing for domain scaling. *arXiv preprint arXiv:1707.02363*.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Lam Monica S. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv:1902.10909*.

Yann N Dauphin, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2013. Zero-shot learning for semantic utterance classification. *arXiv preprint arXiv:1401.0509*.

Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th Annual Symposium on User Interface Software and Technology*, UIST '17.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Xinya Du, Luheng He, Qi Li, Dian Yu, Panupong Pasupat, and Yuan Zhang. 2021. QA-driven zero-shot slot filling with weak supervision pre-training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 654–664, Online. Association for Computational Linguistics.

Ali El-Kahky, Xiaohu Liu, Ruhi Sarikaya, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2014. Extending domain coverage of language understanding systems via intent transfer between domains using knowledge graphs and search query click logs. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4067–4071. IEEE.

Gabor Fuisz, Ivan Vulić, Samuel Gibbons, Inigo Casanueva, and Paweł Budzianowski. 2022. Improved and efficient conversational slot labeling through question answering. *arXiv preprint arXiv:2204.02123*.

Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2022. Zero-shot learning for joint intent and slot labeling. *arXiv preprint arXiv:2212.07922*.

Chih-Wen Goo, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung Chen. 2018. Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757, New Orleans, Louisiana. Association for Computational Linguistics.

Dilek Hakkani-Tür, Malcolm Slaney, Asli Celikyilmaz, and Larry Heck. 2014. Eye gaze for spoken language understanding in multi-modal conversational interactions. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 263–266.

Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Vivian Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association (INTERSPEECH 2016)*. ISCA.

Larry Heck, Dilek Hakkani-Tür, Madhu Chinthakunta, Gokhan Tur, Rukmini Iyer, Partha Parthasarathy, Lisa Stifelman, Elizabeth Shriberg, and Ashley Fidler. 2013. Multi-modal conversational search and browse. In *Proceedings of the First Workshop on Speech, Language and Audio in Multimedia (SLAM 2013)*, pages 96–101.

Larry Heck and Simon Heck. 2020. Zero-shot visual slot filling as question answering. *CoRR*, abs/2011.12340.

Aaron Jaech, Larry Heck, and Mari Ostendorf. 2016. Domain adaptation of recurrent neural networks for natural language understanding. *arXiv:1604.00117*.

Susmit Jha, Sumit Gulwani, Sanjit A Seshia, and Ashish Tiwari. 2010. Oracle-guided component-based program synthesis. In *2010 ACM/IEEE*

32nd International Conference on Software Engineering, volume 1, pages 215–224. IEEE.

Saurabh Kumar, Pararth Shah, Dilek Hakkani-Tur, and Larry Heck. 2017. Federated control with hierarchical multi-agent deep reinforcement learning. In *Conference on Neural Information Processing Systems (NeurIPS), Hierarchical Reinforcement Learning Workshop*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv:1909.11942*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *CoRR*, abs/1706.04115.

Dawei Li, Serafettin Tasci, Shalini Ghosh, Jingwen Zhu, Junting Zhang, and Larry Heck. 2019. Rilod: near real-time incremental learning for object detection at the edge. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 113–126.

Xuefeng Li, Liwen Wang, Guanting Dong, Keqing He, Jinzheng Zhao, Hao Lei, Jiachi Liu, and Weiran Xu. 2023. Generative zero-shot prompt learning for cross-domain slot filling with inverse prompting.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2017. End-to-end optimization of task-oriented dialogue model with deep reinforcement learning. *arXiv preprint arXiv:1711.10712*.

Thomas F. Liu, Mark Craft, Jason Situ, Ersin Yumer, Radomir Mech, and Ranjitha Kumar. 2018. Learning design semantics for mobile apps. In *The 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, pages 569–579.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Shikib Mehri, Yasemin Altun, and Maxine Eskenazi. 2022. LAD: Language models as data for zero-shot dialog. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 595–604, Edinburgh, UK. Association for Computational Linguistics.

Shikib Mehri and Maxine Eskenazi. 2021. GenSF: Simultaneous adaptation of generative pre-trained models and slot filling. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 489–498, Singapore and Online. Association for Computational Linguistics.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2014. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Mahdi Namazifar, Alexandros Papangelis, Gokhan Tur, and Dilek Hakkani-Tür. 2021. Language model is all you need: Natural language understanding as question answering. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7803–7807.

Eda Okur, Shachi H Kumar, Saurav Sahay, and Lama Nachman. 2019. Towards multimodal understanding of passenger-vehicle interactions in autonomous vehicles: Intent/slot recognition utilizing audio-visual data.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. pages 784–789.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8689–8696.

B. Reichman and L. Heck. 2023. Cross-modal dense passage retrieval for outside knowledge visual question answering. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2829–2834, Los Alamitos, CA, USA. IEEE Computer Society.

Benjamin Z. Reichman, Anirudh Sundar, Christopher Richardson, Tamara Zubatiy, Prithwijit Chowdhury, Aaryan Shah, Jack Truxal, Micah Grimes, Dristi Shah, Woo Ju Chee, Saif Punjwani, Atishay Jain, and Larry Heck. 2023. Outside knowledge visual question answering version 2.0. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Ramprasaath R Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry Heck, Dhruv Batra, and Devi Parikh. 2019. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2591–2600.

Pararth Shah, Dilek Hakkani-Tür, and Larry Heck. 2016. Interactive reinforcement learning for task-oriented dialogue management. In *Conference on Neural Information Processing Systems (NIPS), Workshop on Deep Learning for Action and Interaction*.

Guangzhi Sun, Chao Zhang, Ivan Vulić, Paweł Budzianowski, and Philip C Woodland. 2023. Knowledge-aware audio-grounded generative slot filling for limited annotated data. *arXiv preprint arXiv:2307.01764*.

Anirudh Sundar, Christopher Richardson, and Larry Heck. 2024. gtbls: Generating tables from text by conditional question answering.

Anirudh S Sundar and Larry Heck. 2022. Multimodal conversational ai a survey of datasets and approaches. *ACL 2022*, page 131.

Anirudh S. Sundar and Larry Heck. 2023. cTBLS: Augmenting large language models with conversational tables. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pages 59–70, Toronto, Canada. Association for Computational Linguistics.

Gokhan Tur and Renato De Mori, editors. 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*. Wiley.

Gokhan Tur, Dilek Hakkani-Tür, and Larry Heck. 2010. What is left to be understood in atis? In *2010 IEEE Spoken Language Technology Workshop*, pages 19–24. IEEE.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tur, Hakkani-Tür Dilek, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. IEEE.

Nancy Xu, Sam Masling, Michael Du, Giovanni Campagna, Larry Heck, James Landay, and Monica Lam. 2021. Grounding open-domain instructions to automate web support tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1022–1032.

Silei Xu Xu, Sina J. Semnani, Giovanni Campagna, and Monica S. Lam. 2020. Autoqa: From databases to qa semantic parsers with only synthetic training data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

D Yann, G Tur, D Hakkani-Tur, and L Heck. 2014. Zero-shot learning and clustering for semantic utterance classification using deep learning. In *International Conference on Learning Representations (cited on page 28)*.

Heming Zhang, Shalini Ghosh, Larry Heck, Stephen Walsh, Junting Zhang, Jie Zhang, and C-C Jay Kuo. 2019. Generative visual dialogue system via weighted likelihood estimation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 1025–1031.

Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. 2020. Class-incremental learning via deep model consolidation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1131–1140.

Wenting Zhao, Konstantine Arkoudas, Weiqi Sun, and Claire Cardie. 2022. Compositional task-oriented parsing as abstractive question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4418–4427, Seattle, United States. Association for Computational Linguistics.

6. Appendix

This section outlines the applications used to collect the mForms dataset. The slot schemas for each form are shown in Table 6.

Vehicle Logger: The Vehicle Logger App shown in Figure 6 is a popular tool to create, share, and report vehicle log books for mileage, fuel expenses, and tax purposes. As previously described, the visual GUI elements of the Vehicle Logger App include Text fields (e.g., Odometer Value), Radio Buttons (e.g., Business, Personal, Other), and Text Buttons (e.g., Track distance with GPS). Referring to Table 1, 850 utterances were collected with annotations according to 10 slot types.

United Airlines: The United Airlines flight search App shown in Figure 7 is used to find flights according to travel plans and preferences. The GUI elements include simple text fields, tab buttons, and search buttons as well as more visually-oriented icons such as the user's current location (icons on the right-most column) and an icon to swap departure and arrival airports. 850 utterances were collected with annotations according to 6 slot types.

Trip Advisor: Finally, Figure 8 shows the Trip Advisor App. This App serves many purposes including booking a table at restaurants as well as comparing prices when booking flights and hotels. The portion of the App used for this study focused on hotel room booking. Much of the App screen shown in the Figure contains visually oriented icons such as the symbol for people (in this case, showing 2 people) and a bed (1 bed in the room). The Trip Advisor dataset has 803 utterances with annotations according to 6 slots.
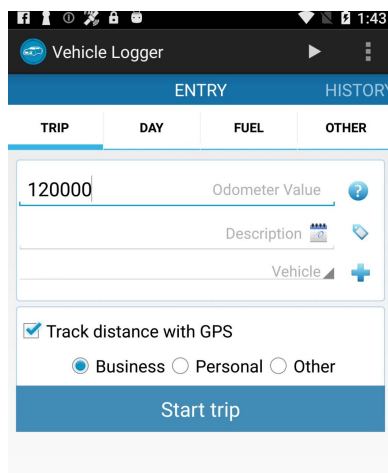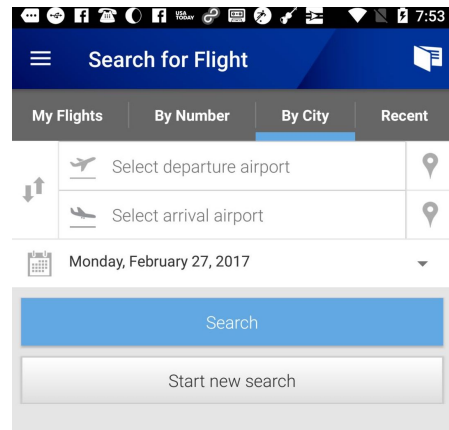

Figure 7: United flight search Application

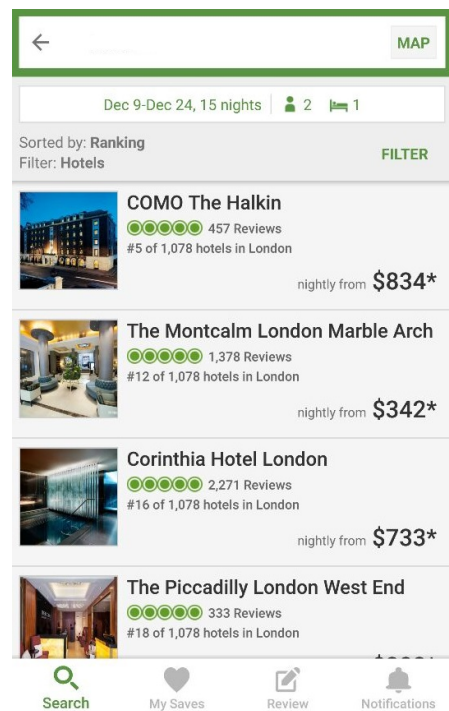Figure 8: Trip Advisor Application

Figure 6: Vehicle Logger Application

Visual App	Slot descriptions
2*Vehicle Logger	fuel cost, fuel added, trip description, gps tracking, start logging, date, odometer value, trip type, entry, vehicle
United	arrival airport, departure airport, travel dates, search, switch/swap airports
2*ATIS form-filling	aircraft code, airline code, airline name, airport code, airport name, arrival date (relative), arrival date (day name), arrival date (day number), arrival date (month name), etc
Trip Advisor	number of beds, date range, filter by price, filter by rating, number of nights, number of people

Table 6: Slot schema / descriptions used for the mForms tagger for each domain
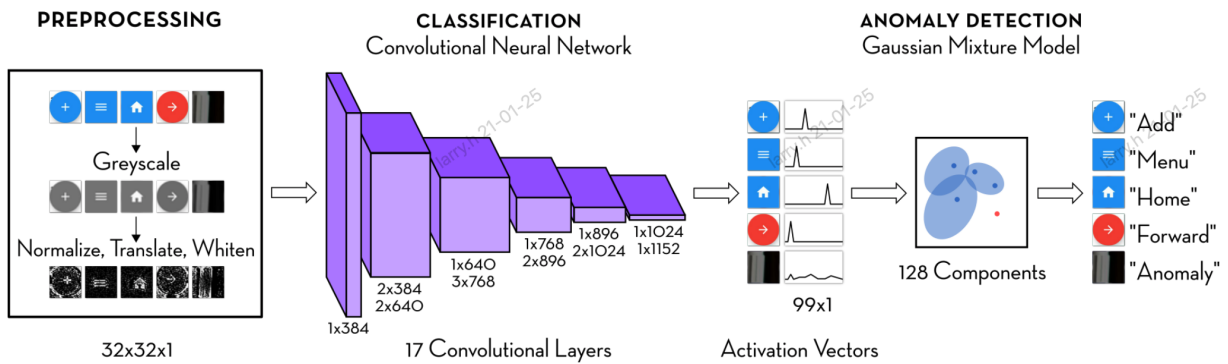

Figure 9: Computer Vision classification of GUI visual elements (Liu et al. 2018)