Modeling Intent, Dialog Policies and Response Adaptation
for Goal-Oriented Interactions

Saurav Sahay, Shachi H Kumar, Eda Okur, Haroon Syed, Lama Nachman

Intel Labs, Anticipatory Computing Lab, USA

{saurav.sahay, shachi.h.kumar, eda.okur, haroon.m.syed, lama.nachman}
@intel.com

Abstract

Building a machine learning driven spoken dialog system for goal-oriented interactions involves careful design of intents and data collection along with development of intent recognition models and dialog policy learning algorithms. The models should be robust enough to handle various user distractions during the interaction flow and should steer the user back into an engaging interaction for successful completion of the interaction. In this work, we have designed a goal-oriented interaction system where children can engage with agents for a series of interactions involving 'Meet & Greet' and 'Simon Says' game play. We have explored various feature extractors and models for improved intent recognition and looked at leveraging previous user and system interactions in novel ways with attention models. We have also looked at dialog adaptation methods for entrained response selection. Our bootstrapped models from limited training data perform better than many baseline approaches we have looked at for intent recognition and dialog action prediction.

1 Introduction

Language technologies have benefited from recent progress in AI and Machine Learning. There have been major advancements in spoken-language understanding (Mikolov et al., 2013; Mesnil et al., 2015). Machine-learning approaches to dialog management have brought improved performance compared to traditional handcrafted approaches by enabling systems to learn optimal dialog strategies from data (Paek and Pieraccini, 2008; Bangalore et al., 2008). With availability of large amounts of data and advancements in deep learning research, end-to-end trained systems have also shown to produce state of the art results in both open-ended (Dodge et al., 2015) and goal-oriented

applications (Bordes et al., 2016) in the research community.

With the emergence of reliable ASR and TTS systems, we are now seeing platforms such as Alexa and Google Home and a plethora of domain and task-based dialog agents that allow users to take specific actions via spoken interface-based systems. Microsoft Research released Language Understanding Intelligent Service (LUIS) (Williams et al., 2015a,b), which helps software developers create cloud-based, machine-learning powered, language-understanding models for specific application domains. Google Dialogflow is an SDS platform for quick development of various conversational agents that can be deployed on various platforms such as Amazon Alexa, Google Home and several others. Systems like Google's Dialogflow offer mechanisms such as explicit context management on linear and non-linear dialog flows to manage the conversations. The developer can attach input and output context states explicitly on various intents and create if-then-else flows via context variables. To go from explicit context management to implicit/automatic context management in SDS, probabilistic and neural network based systems are emerging in the research and development community (Bocklisch et al., 2017; Burtsev et al., 2018; Ultes et al., 2017).

Most dialog systems are categorized as either chatbots or task-oriented where chatbots are open-ended to allow generic conversations and the task-oriented system is designed for a particular task and set up to have short conversations (Jurafsky and Martin, 2018 (Online). Goal-oriented Interaction Systems are somewhat midway between the two and should support longer duration interactions having various tasks to fulfill as well as support some non-task interactions.

We are developing a goal-oriented multimodal conversational system that engages 5 to 7 years old

children in concise interaction modules (Anderson et al., 2018). The overall goal of the system is to engage children in multimodal experiences for playful and learning oriented interactions.

Our application consists of interactions where children get introduced to the agent and they can play a simplified version of 'Simon Says' game with the agent. The dialog manager ingests verbal and non-verbal communication via the Natural Language Understanding (NLU) component (entities and intents) and other engines that process vision and audio information (faces, pose, gesture, events and actions) and generates sequential actions for utterances and non-verbal events. We describe the NLU, Dialog Manager (DM) and Dialog Adaptation modules in this work as shown in Figure 1. We build our NLU and DM based models on top of the Rasa framework (Bocklisch et al., 2017). We enrich the NLU Intent Recognition model in Rasa by adding additional features to the model. Typical goal-oriented dialog managers are modeled as sequential decision making problems where optimal interaction policies are learned from a large number of user interactions via reinforcement learning approaches (Shah et al., 2016; Liu et al., 2017; Su et al., 2017; Levin and Pieraccini, 1997; Cuayáhuitl, 2017; Dhingra et al., 2016). Building such systems to support children interactions is particularly difficult and we use a supervised learning approach using some initial training data collected via Amazon Mechanical Turk (AMT) to bootstrap the agents. Dialog Adaptation has been explored as part of parametric generation (Mairesse and Walker, 2007) as well as end-to-end NLG for generating contextually appropriate responses (Dušek and Jurčíček, 2016). We look at parametric generation methods and develop a simple ML classifier for Dialog Adaptation.
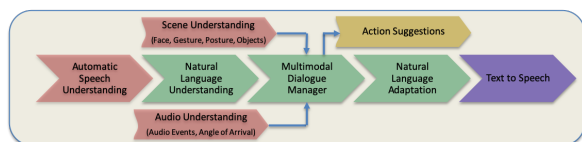

Figure 1: Multimodal Spoken Dialog System

2 Data Collection

Guided by User Experience (UX) research and Wizard of Oz user studies involving kids, we have designed sample interactions between the agent and the kid (Figure 2) for the 'Meet & Greet' and 'Simon Says' interactions. The left half of Fig. 2 shows the task-oriented dialog flow. However, in a real system, one could imagine a lot of non-task-oriented or non-cooperative dialog involved. Especially in a dialog involving young kids, there could be a lot of chit-chat style conversations on things such as preferences or likes/dislikes of kids, a description of how the day went at school, or even the kids asking several questions about the agent itself. As shown in the right half of the Fig. 2, these can be categorized as either simple chit-chat or some unexpected utterances, as well as complaints or requests for help. To support our application, which includes a mixture of task and non-task-oriented conversations, we collect data via AMT for two types of interactions: 'Meet & Greet', and 'Simon Says', between the agent and the kid. We requested the turkers to provide us with a dialog, with agent on one side and kid on the other by providing sample interaction flows as shown in Fig. 2.

We collected 80 dialogs in total (65 stories for training, and 15 for test) for the two interactions. After the annotation process, we observed 48 user intents (including verbal intents and physical activities for the 'Simon Says' game), as well as 48 distinct agent actions. Note that for our NLU models, 26 distinct verbal intents are observed in the dataset, where 16 of them are categorized as goal-oriented (i.e., related to our 'Meet & Greet' and 'Simon Says' scenarios), and the remaining 10 are non-goal-oriented (e.g., chit-chat or out-of-scope). Further details of our NLU and dialog datasets can be found in Table 1 and Table 2, respectively.
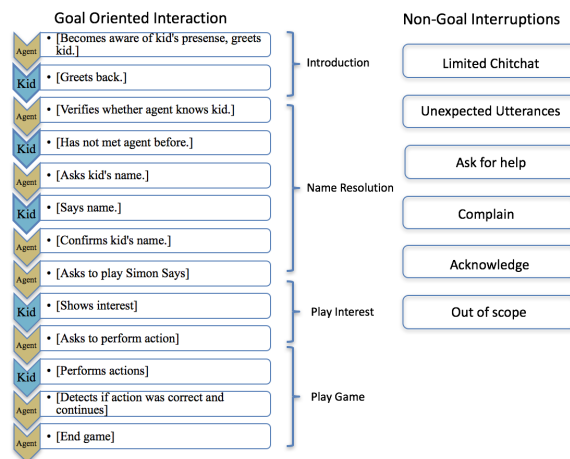

Figure 2: Domain of Application

	Training	Test
# of intents (distinct)	26	26
goal-oriented	16	16
non-goal-oriented	10	10
total # of samples (utterances)	779	288
total # of tokens	3850	1465
mean # of samples per intent	29.96	11.08
mean # of tokens per sample	4.94	5.09

Table 1: NLU Dataset Statistics

	Training			Test		
# of dialogs	65			15		
Meet & Greet	49			12		
Simon Says	16			3		
	Kid	Agent	All	Kid	Agent	All
# of intents/actions	48	48	96	28	28	56
goal-oriented	39	41	80	19	22	41
non-goal-oriented	9	7	16	9	6	15
# of turns	441	560	1001	97	112	209
goal-oriented	374	501	875	63	84	147
non-goal-oriented	67	59	126	34	28	62
# of turns per dialog	6.8	8.6	15.4	6.5	7.5	14.0

Table 2: Dialog Dataset Statistics

3 Models and Architecture

In this section, we describe the architectures we developed for the NLU, DM and Dialog Adaptation modules of the spoken dialog system pipeline.

3.1 NLU/Intent Understanding

The NLU module processes the input utterance to determine the user intents and entities of interest.

3.1.1 Intent Classifier

We use an intent classifier based on supervised embeddings provided as part of the Rasa framework (Bocklisch et al., 2017). This embedding-based intent classifier is based on ideas from the StarSpace algorithm (Wu et al., 2017) and embeds user utterances and intents into the same vector space. These embeddings are trained by maximizing the similarity between them. We also adapt sequence models and Transformer networks[1] to work with various features for developing our models.

3.1.2 Features and models for the NLU module

We utilize and extend the Rasa NLU module by adding various textual features and models to improve the performance of the intent classifier.

Textual features: We used text features such as number of words, first word, last word, bigrams, dependencies such as 1st/2nd/3rd person subject, inverted subject-verb order and imperative verbs, morphology features, hand constructed word lists,

'wh' words, top n words, and many more. We add about 580 such textual features to our custom feature extractor.

Speech Act features: Verbal Response Modes (VRM) is a principled taxonomy of speech acts that can be used to classify literal and pragmatic meaning within utterances (Lampert et al., 2006). Utterances are classified into disjoint sets comprising Question, Disclosure, Edification, Advisement, Acknowledgement, Reflection, Interpretation and Confirmation according to this model[2]. The classifier (Sahay et al., 2011) also used the above text features for modeling the task and the top features in this classification task were domain independent features such as '?', length of words, 'you', 'i','okay', 'well', etc.

Universal Sentence Embeddings: Universal Sentence Encoders (Cer et al., 2018) encode sentences into high dimensional vectors that has shown success in a variety of NLP tasks. We use the encoder model trained using a Transformer encoder[3] to generate fixed length vectors as features for the NLU module in our pipeline. The motivation for using this model is to hope to recognize short utterances with similar meaning where the word level vectors may not provide enough information for correct intent classification.

Sequence Models: Long Short Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) and Bidirectional LSTMs (Schuster and Paliwal, 1997) can capture patterns and dependencies in sequential data using their memory gates and can robustly encode information. We use LSTM and BiLSTM layers to generate representations that are used in place of the fully connected embedding layer in the baseline model.

Transformers: (Vaswani et al., 2017) proposed a novel sequence-to-sequence network, the Transformer, entirely based on attention mechanism. The performance of Transformer model has generally surpassed RNN-based models and achieved better results in various NLP tasks. We use rich representations from the transformer model as an extension to the baseline model.

Bidirectional Encoder Representations from Transformers (BERT): Bidirectional Encoder Representations from Transformers, BERT (Devlin et al., 2018) represents one of the latest de-

---

[1]https://github.com/RasaHQ/rasa/pull/4098

[2]For the classifier, Disclosure and Edification classes were combined into one class

[3]https://tfhub.dev/google/universal-sentence-encoder-large/3

velopments in pre-trained language representation and has shown strong performance in several NLP tasks. We use pre-trained BERT model based features to generate representations for our dataset and use this in our NLU pipeline.

3.2 Dialog State Tracking

The task of a Dialog Manager is to take the current state of the dialog context and decide the next action for the dialog agent by using some policy of execution. Policies that are learnt from actual conversational data either use Markov Decision Processes (MDP) or Memory augmented Neural Networks. Memory augmented networks can update the context of the dialog by storing and retrieving the dialog state information. The dialog states and past system actions can be encoded as domain knowledge in the network to encode dialog manager based rules using Hybrid Code Networks (HCN) (Williams et al., 2017). One such dialog management policy that combines memory operations with domain knowledge embedding is the Recurrent Embedding Domain Policy (REDP) (Vlasov et al., 2018). This policy represents the dialog state embedding as a recurrent network with explicit management of the user memory and system memory modules. The user memory is used to attend to previous user intents and the system memory is used to attend to previous system actions. REDP uses Bahdanau Attention scoring to attend to previous user utterances (user memory) and memory address by location as developed in Neural Turing Machines (NTM) for system memory. With NTM, the previous system actions can be copied directly into the final dialog state representations to help recover from a sequence of uncooperative user utterances. REDP learns embeddings for dialog and system actions in the same space using ideas from the StarSpace algorithm (Wu et al., 2017) and uses these embeddings to rank the similarity between the recurrent embedding and the system action. Figure 3 shows the overall architecture of the dialog state tracker.

Our application handles multimodal streams of high frequency non-verbal input such as person recognition via face and speaker identification, gestures, pose and audio events. We pass on the information via separate modules to the dialog manager (bypassing the NLU) as relevant intents for goal-oriented interaction.
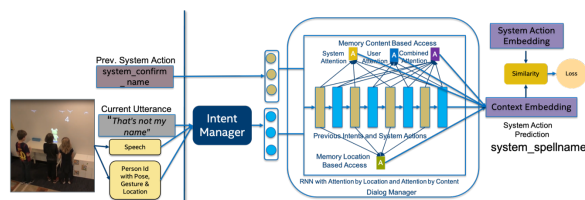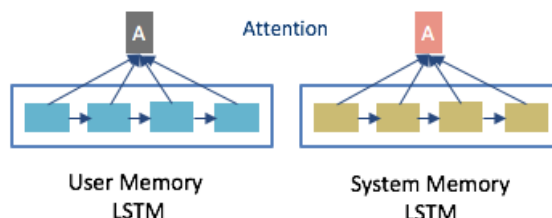

Figure 3: Dialog State Tracking Architecture

Figure 4: User Memory and System Memory

3.2.1 User and System Memories

Attention-based models (Bahdanau et al., 2014) can dynamically retrieve relevant pieces of information via selective reading through a relatively simple matching operation. In the REDP baseline architecture, separate mechanisms are used for attending to past user utterances and past system actions. While the system memory block helps the agent recover from uncooperative dialog by attending to and copying past system actions (using NTMs), the user memory mechanism uses Bahdanau Attention based matching to attend to relevant parts of past user utterances and enhance the user intent signal in the LSTM.

3.2.2 Fusion of User and System Memory

In this work, we capture the previous user-system interactions into the recurrent architecture by fusion of the signals from the user inputs and system actions. We hypothesize that attending to previous combinations of user utterances and system actions can help the bot choose the right action by directly leveraging multiple discrete views of the dialog context information. This may be useful in contexts involving multi-party interactions. Agents can also benefit from discrete attention in situations where deviations from the task-oriented dialog flow can lead to small multi-turn interaction where the context of the dialog (combinations of previous user utterances and responses) is crucial.

Figure 4 shows the memory units based on previous user intents and system actions. Figure 5 shows the fused memory unit obtained using the

dot product of user memory and system memory. It computes the product of all possible interactions between the intent features and system action features to obtain a memory of size [*user embedding x system action embedding*]. This results in an enhanced representation of the dialog embedding vector (2 below). This dialog embedding vector is further enhanced using NTM based copying of the relevant previous system action as described in (Vlasov et al., 2018).

We incorporate another fusion technique as described in (Zadeh et al., 2017) where one larger memory component explicitly represents user memory, system memory and the interactions between the two (3 below).

Figure 6: User Memory x System Memory Fusion

the fused user-system memory blocks. Configuration 6 above is also conceptually shown in Figure 7 with separate attention blocks to user memory, system memory and the combined user and system memory blocks.
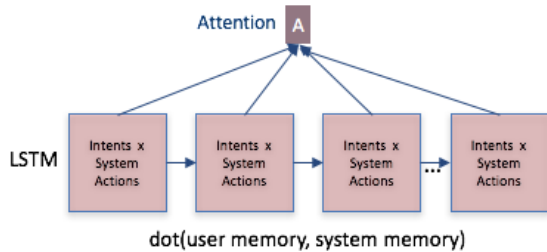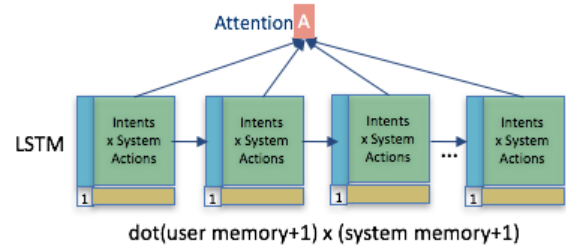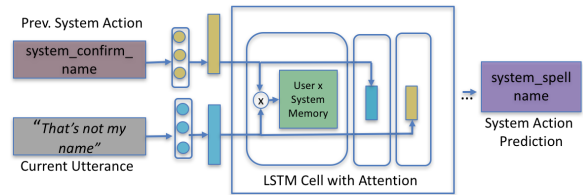

Figure 5: Capturing interactions between intents and system actions

We create the following combinations of the user and system memory blocks as part of the LSTM cell for attending to these memories for computing the relevant system action:

1. Concatenation of User and System memories (Single Memory Unit)

2. Tensor Dot of User and System memories (Single Memory Unit)

3. Tensor Fusion of User and System memories (Single Memory Unit)

4. Separate Attention on User and System memories (Two Memory Units)

5. Separate Attention on User memory and Tensor Dot (Two Memory Units)

6. Separate Attention on User memory, System memory and Tensor Dot (Three Memory Units)

7. Separate Attention on User memory, System memory and Tensor Fusion (Three Memory Units)

The configurations in the list above are a combination of the user memory, system memory and

Figure 7: Simplified View of Contextual Attention

3.3 Response Adaptation

Children adapt to syntax and words used by their dialog partner more frequently than adults (Nilsenová and Nolting, 2010). This phenomenon is called Entrainment and generally applies to copying the conversational partner's attributes related to phonetics (Pardo, 2006), syntax (Reitter et al., 2006), linguistic style (Niederhoffer and Pennebaker, 2002), postures, facial expressions (L. Chartrand and A. Bargh, 1999), etc. It has also been linked to overall task success (Reitter and Moore, 2007) and social factors (Ireland et al., 2011). In this work, we explore lexical and syntactic adaptation, by using the similar referring expressions or sentence structure. SDS could pick a response at random from a list of responses for actions to match the predicted dialog state. In our work, we use the response adaptation module to score the responses and choose the best response instead of picking any random response. Figure 8 shows our architecture for response adaptation.

4 Experiments and Results

In this section we present and discuss the results for the NLU and DST experiments based on the architectural explorations explained in Section 3.
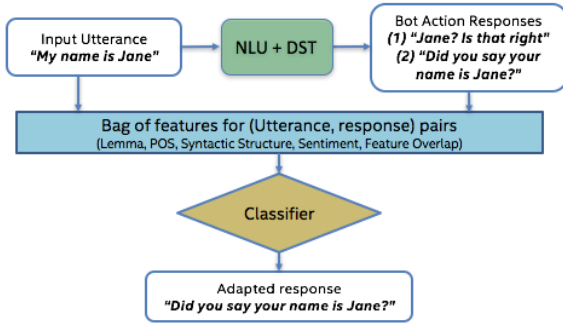
Figure 8: Response Adaptation

	Prec	Rec	F1
Baseline (StarSpace)	0.74	0.65	0.66
Baseline + LSTM	0.75	0.70	0.70
Baseline + BiLSTM	0.76	0.70	0.70
Baseline + Transformer	0.72	0.70	0.68
Baseline + BERT + BiLSTM	0.84	0.81	0.81
Baseline + BERT + Transformer	**0.87**	**0.82**	**0.83**
Baseline + SA	0.74	0.68	0.68
Baseline + USE	0.76	0.71	0.72
Baseline + BERT	0.81	0.77	0.77
Baseline + USE + SA	0.76	0.72	0.73
Baseline + USE + BERT	0.82	0.76	0.77
Baseline + USE + SA + BERT	0.83	0.78	0.78

Table 3: NLU Ablation Experiments on Meet & Greet and Simon Says Dataset

4.1 NLU Experiments

To evaluate the feature explorations for the NLU model, we present an ablation study on the LSTM, BiLSTM, Transformer, USE, text + speech acts features (SA) and BERT feature additions presented in Section 3.1. We use the embedding intent classifier as the baseline for the NLU experiments. For sequence models and Transformer model, we use word level features instead of the sentence-level features. Table 3 shows the precision, recall and F1-scores for the overall intent identification using different features.

The performance of the baseline model with the text + speech act features added is very similar to the baseline. Adding USE to the baseline improves the scores and this can be explained from the fact that USE, which is a transformer based model, generates better representations compared to individual text/speech act features. Consistent with the fact that BERT has shown a lot of success in several NLP tasks recently, we notice that adding BERT representations alone to the baseline improves the performance significantly. On the word-level features, we observe that while LSTM, BiLSTM and Transformer representations show slight improvement, adding BERT along with these shows a significant performance increase. In general, from Table 3, we can conclude that the speech act features have a very minor impact on the performance. The results show that adding BERT with the Transformer model helps achieving best performance on the dataset.

Table 4 presents some qualitative comparison of the baseline model with the text + speech acts (B+SA), USE (B+USE) and BERT (B+BERT) features against the ground truth labels (GT). From the table, we see that a short utterance such as 'i am okay' is labeled as 'mynameis'

by the B+SA model. We believe that this can be attributed to the text features that look at words/phrases such as 'i am', and the 'mynameis' intent would usually start with 'i am'. B+USE model predicts the correct label for this utterance. Although B+BERT model assigns the incorrect label, the prediction is semantically close to the GT. We again observe that a very short phrase such as 'oh shit' is classified incorrectly by both the B+SA and B+BERT models. We believe that for very short utterances, the USE model generates better representations as compared to BERT, and hence can produce more meaningful predictions.

An interesting point to observe is that, for utterances such as 'that ain't me Oscar' or 'are you Alexa?', the BERT feature based model associates them with the 'deny' and 'useraskname' intents, respectively. Although these intents are wrongly identified, they are semantically very close to the ground truth labels 'wrongname' and 'askaboutbot'. While the B+SA model tends to generate incorrect predictions for challenging examples, the B+USE model classifies 'are you Alexa?' as out-of-scope. Longer utterances such as 'where is the nearest shopping mall' are well handled by the BERT model, while the other models fail. We can conclude that the USE model could better handle very short sentences, and the BERT model performs better on the longer ones.

4.2 Dialog State Tracking Experiments

We investigate the role of single vs. multiple memories for attention as well as the impact of system memory and user memory fusion with the policy explorations. In Figure 9, we compare the results of the baseline REDP policy from (Vlasov

Kid's utterance	B+SA	B+USE	B+BERT	GT
'i am okay'	mynameis	userdoinggood	askhowdoing	userdoinggood
'oh shit'	askhowdoing	usermissedIt	affirm	usermissedIt
'are you Alexa?'	askhowdoing	outofscope	useraskname	askaboutbot
'that ain't me Oscar'	outofscope	mynameis	deny	wrongname
'where is the nearest shopping mall'	nextstep	useriamSick	outofscope	outofscope

Table 4: Qualitative analysis of baseline with USE+SA as features vs baseline with BERT

et al., 2018) with our policy changes on the dataset used by the authors that contain uncooperative and cooperative dialogs from hotel and restaurant domain. We use the same test set from the hotel domain and use a combination of cooperative and uncooperative dialogs from both hotel and restaurant domain for the training set. We divide the training set into 7 splits with 0, 5, 25, 50, 70, 90, 95 percent exclusion in the number of dialog stories in the domains. The baseline policy from (Vlasov et al., 2018) applies Bahdanau Attention scoring to the history of user utterances only. The policy does not explore attending to previous system actions or combinations of those for updating the RNN states as part of Bahdanau Attention scoring. In our experiments, we reuse the NTM based memory copying mechanism for system actions but explore additional effects of leveraging previous system actions and their combinations with the previous user intents. We see that using separate attention blocks on user memory, system memory and the combined memory using their dot product interactions help achieve slightly improved performance on this dataset. We can see some advantages in the case of very limited training data (when the agent cannot perhaps copy previous system actions) as well as in the case of full training set, where we

see a slightly higher number of correct test stories with other policies. Further investigation is needed to understand if the proposed policy changes in REDP would always benefit in certain scenarios. We also try to investigate the comparison between using a single (larger) memory attention vs. using multiple memory attentions. For example, as shown in Figure 9, 3 policy changes perform better than the baseline policy, all of which use separate memory blocks and all of these attend to the fused interaction representation of user and system memory.

Figure 10 shows the action prediction F1-scores for our 'Meet & Greet and Simon Says' dataset. Our test stories were collected using AMT and we allowed the bot and user to engage in any number of utterances and actions before passing control to the other. Since we also did not also impose any particular sequence in the game play, we didn't expect to get all action predictions correct in any one of test stories. We show the F1-scores for action predictions for the test stories varying the size of the training set. The more training data the agent has seen, more user utterance and system action interactions it has seen capturing application regularities, therefore we can hope to see improved
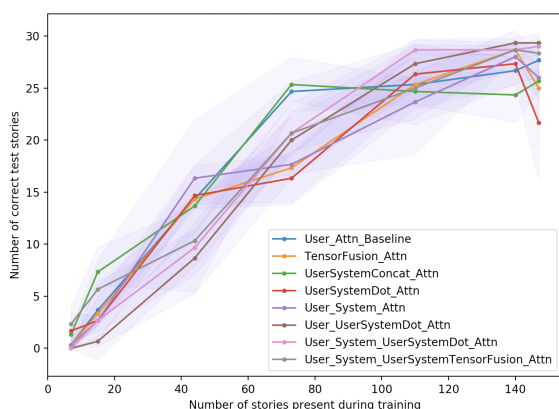

Figure 9: Performance of Models with RNN Attention over User and System Memory configurations with varying Training Sizes
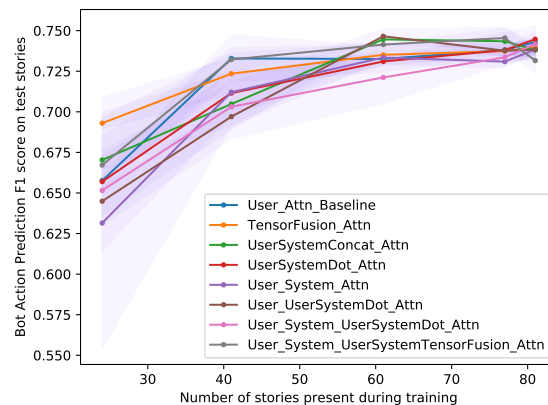

Figure 10: Action Prediction F1-score of Models with RNN Attention over User and System Memory configurations with varying Training Sizes

bot performance on unseen test sets with multiple memories and fusion configurations of attention units. From Figure 10, we can only say that there is a lot of variance in predictions with lesser training data, general trend for all these policies is get better with more training data. We see that the overall best F1-score is achieved with 75% of the training data with two separate attentions, one for user memory and another for the user-system fusion interactions.

4.3 Dialog Adaptation

For generating contextually appropriate responses, we collected multiple responses for bot utterances and trained ML models for selecting the most appropriate response from the list of responses. The goal of the classifier was to match the syntactic and linguistic style of the speaker. For this, we used 60 dialogs, with 32400 instances (2753 positive instances, 29647 negative instances) and 243 features. We created positive and negative instances automatically using feature overlap counts between the context dialog and the responses to be adapted. For feature generation, we extracted lemmas, Parts of Speech, Syntactic Structures and Sentiment Polarities using Stanford CoreNLP suite (Manning et al., 2014).
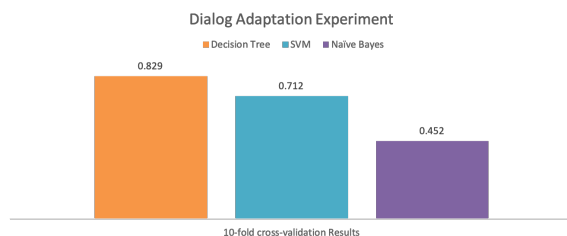
Figure 11: Dialog Adaptation

Figure 11 shows the cross-validation performance of the classifiers on the dataset. We use the Decision Tree based classifier for response adaptation in the spoken dialog system. Figure 12 shows a couple of response adaptation examples along with the positive and negative classification results for the two context utterances.

5 Conclusions & Future Work

We report preliminary explorations and results for our data driven spoken dialog system development for the multimodal 'Meet & Greet and Simon Says' goal-oriented application. The application involves phases of interactions for in-
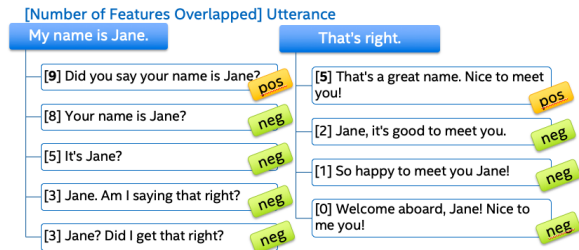
Figure 12: Dialog Adaptation Examples

troduction, name resolution, game related interaction and actual game play involving children. We collect NLU and Dialog Data for our application using AMT, and manually identify non-goal-oriented intents and design interactions to include various non-goal-oriented or 'uncooperative' paths in the interaction. We have used and extended the Rasa NLU module and Rasa Core module for Dialog Management. Our application involves five to seven year-old children communicating with agents and we have seen from data that many children use very short utterances. In order to have a robust NLU, we have explored the use of lexical, syntactic and speech act related features (SA features), Universal Sentence Encoders as well as BERT embeddings for the embedding-based intent classifier which is a part of the Rasa NLU stack. We see the largest improvement in the NLU performance using the pre-trained BERT features and the Transformer model. For Dialog State Tracking, we extended the REDP policy by including different configurations of User and System Memory for RNN based Attention. We looked at a method for Single Memory Unit Tensor Fusion for combining User Memory, System Memory and tensor fused representation of User and System Memory. We explore other multiple memory unit configurations for RNN based Attention on history of User Intents, System Actions and their combinations. We saw improvements over the REDP baseline policy for the hotel and restaurant domain dataset as well as the 'Meet & Greet and Simon Says' dataset. We also explored Response Selection from the list of response templates as an Adaptation Classification problem using features such as Lemma/POS, Syntactic feature overlap and Sentiment of the response. As part of future work, we plan to extend the NLU and DM based models to include multimodal information in the pipeline.

Acknowledgments

References

Glen J. Anderson, Selvakumar Panneer, Meng Shi, Carl S. Marshall, Ankur Agrawal, Rebecca Chierichetti, Giuseppe Raffa, John Sherry, Daria Loi, and Lenitra Megail Durham. 2018. Kid space: Interactive learning in a smart environment. In *Proceedings of the Group Interaction Frontiers in Technology Workshop, GIFT@ICMI 2018, Boulder, CO, USA, October 16, 2018*, pages 8:1–8:9. ACM.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Srinivas Bangalore, Giuseppe Di Fabbrizio, and Amanda Stent. 2008. Learning the structure of task-driven human–human dialogs. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1249–1259.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. 2018. DeepPavlov: Open-source library for dialogue systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127, Melbourne, Australia. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 169–174. Association for Computational Linguistics.

Heriberto Cuayáhuitl. 2017. Simpleds: A simple deep reinforcement learning dialogue system. In *Dialogues with Social Robots*, pages 109–118. Springer.

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

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2016. Towards end-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777*.

Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander Miller, Arthur Szlam, and Jason Weston. 2015. Evaluating prerequisite qualities for learning end-to-end dialog systems. *arXiv preprint arXiv:1511.06931*.

Ondřej Dušek and Filip Jurčíček. 2016. A context-aware natural language generator for dialogue systems. *arXiv preprint arXiv:1608.07076*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Molly E. Ireland, Richard B. Slatcher, Paul W. Eastwick, Lauren E. Scissors, Eli J. Finkel, and James W. Pennebaker. 2011. Language style matching predicts relationship initiation and stability. *Psychological Science*, 22(1):39–44. PMID: 21149854.

Daniel Jurafsky and James H. Martin. 2018 (Online). *Ch 24: Dialog Systems and Chatbots. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3rd (draft) edition. Prentice Hall PTR, Upper Saddle River, NJ, USA.

Tanya L. Chartrand and John A. Bargh. 1999. The chameleon effect: The perception-behavior link and social interaction. journal of personality social psychology, 76(6): 893-910. *Journal of personality and social psychology*, 76:893–910.

Andrew Lampert, Robert Dale, and Cécile Paris. 2006. Classifying speech acts using verbal response modes. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 34–41.

Esther Levin and Roberto Pieraccini. 1997. A stochastic model of computer-human interaction for learning dialogue strategies. In *Fifth European Conference on Speech Communication and Technology*.

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. In *NIPS Workshop on Conversational AI*.

François Mairesse and Marilyn Walker. 2007. Person-age: Personality generation for dialogue. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 496–503.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David Mc-Closky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Kate G. Niederhoffer and James W. Pennebaker. 2002. Linguistic style matching in social interaction. *Journal of Language and Social Psychology*, 21(4):337–360.

Marie Nilsenová and Palesa Nolting. 2010. Linguistic adaptation in semi-natural dialogues: Age comparison. In *Text, Speech and Dialogue*, pages 531–538, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tim Paek and Roberto Pieraccini. 2008. Automating spoken dialogue management design using machine learning: An industry perspective. *Speech communication*, 50(8-9):716–729.

Jennifer S. Pardo. 2006. On phonetic convergence during conversational interaction. *The Journal of the Acoustical Society of America*, 119(4):2382–2393.

David Reitter and Johanna D. Moore. 2007. Predicting success in dialogue. In *ACL 2007 - Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 808–815.

David Reitter, Johanna D. Moore, and Frank Keller. 2006. Priming of syntactic rules in task-oriented dialogue and spontaneous conversation. In *In Proceedings of the 28th Annual Conference of the Cognitive Science Society*.

Saurav Sahay, Hua Ai, and Ashwin Ram. 2011. Intentional analysis of medical conversations for community engagement. In *Twenty-Fourth International FLAIRS Conference*.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *Trans. Sig. Proc.*, 45(11):2673–2681.

Pararth Shah, Dilek Hakkani-Tur, and Larry Heck. 2016. Interactive reinforcement learning for task-oriented dialogue management.

Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 147–157, Saarbrücken, Germany. Association for Computational Linguistics.

Stefan Ultes, Lina M. Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, and Steve Young. 2017. PyDial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Vladimir Vlasov, Akela Drissner-Schmid, and Alan Nichol. 2018. Few-shot generalization across dialogue tasks. *arXiv preprint arXiv:1811.11707*.

Jason D. Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *CoRR*, abs/1702.03274.

Jason D Williams, Eslam Kamal, Mokhtar Ashour, Hani Amr, Jessica Miller, and Geoff Zweig. 2015a. Fast and easy language understanding for dialog systems with microsoft language understanding intelligent service (luis). In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 159–161.

Jason D Williams, Nobal B Niraula, Pradeep Dasigi, Aparna Lakshmiratan, Carlos Garcia Jurado Suarez, Mouni Reddy, and Geoff Zweig. 2015b. Rapidly scaling dialog systems with interactive learning. In *Natural Language Dialog Systems and Intelligent Assistants*, pages 1–13. Springer.

L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston. 2017. Starspace: Embed all the things! *arXiv preprint arXiv:1709.03856*.

Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*.