Final Report
Persona-Based Interactive Agent

Team	Advisors
Alexander J. Orona Martin Malasky	Amarnath Gupta Matthew Fisher

Abstract

Creating agents that closely mimic human conversational patterns has been a major goal of research and industry for decades. Recent advances in recurrent and transformer deep learning architectures have made significant progress toward open domain chatbots that capture the highly contextual nature of conversations. In the last several years, researchers have turned to personas as one way of improving dialogue quality.

We test current methods for building persona-based conversational agents by building a chatbot of a popular character (Rachel from the TV show Friends). Our aim is to: 1) deliver an engaging Friends experience to fans of the show, 2) assess chatbot deep learning architectures, and 3) establish the limits of contemporary approaches and potential areas for additional research.

Introduction and Question Formulation

Recent advances in neural networks have led to notable improvements in interactive dialogue agents (chatbots). For open domain chatbots, this has led to applications as diverse as entertainment to the management of grief and loss (i.e. Eugenia Kuyda's digital memorial to Roman Mazurenko). The range of applications is expected to expand as the human-like quality of dialogue improves. Though research in this field has been ongoing for many years, the introduction of more sophisticated neural network architectures and hardware to effectively train them has enabled machine learning techniques (versus, for example, hardcoding responses) that perform well in real applications.

Context has become a defining factor in improving performance. Li (2016) and others have argued that one aspect limiting performance is the use of training datasets from disparate speakers and the absence of centrally modelled persona. This makes sense in supervised chatbot training: training on a faceless mass of speakers should result in general, consensual agents. Whereas human discourse is highly contextual and grounded in idiosyncratic qualities of a human speaker, their own experiences, histories, affects and social relationships, chatbots have in general not been able to replicate this. If "man is an animal suspended in webs of significance he himself has spun" (Geertz 1973), then what can reasonably expect of an interactive agent that aims to emulate a person in conversation, but which has been trained on essentially impersonal data?

A persona-based approach – by which we mean an approach that seeks to emulate a specific person – may be the most promising method of creating a reasonably good conversational agent. We are interested in the implications of persona-based interactive agents because of the unique areas of application beyond the typical industry considerations mentioned above (e.g. 'customer service'). The chief contribution of this project is toward a general methodology in constructing engaging, persona-based chatbots. The place where development and validation can occur with the least risk is in entertainment, gaming and marketing. It could also be used in creative industries such as screenwriting, theatre and literature to build chatbot models on fictitious characters.

Project Goals

Our project goals were to:

Improving fan engagement: How long will users stay engaged with our chatbot (measured in number of lines in dialog rather than time)?

Insights in evaluation metrics: How do current measures of performance (Perplexity, Bleu, ADEM) compare to real-world feedback? What can be said about these metrics in chatbot evaluation?

Better methods: Does recent work in deep learning improve chatbot performance metrics and real world feedback?

Relevant Work

Recent advances in deep learning architectures have renewed interest in conversational agents such as chatbots (Chen et. al, 2018; Gatt and Krahmer 2018: 138; Young et al. 2018). In comparison to an earlier era of hard-coded response systems, agents build using Sequence2Sequence/recurrent neural networks and universal transformers are able to generate novel responses outside of a constrained domain (Vinyals and Le 2015; Wolf et. al 2019). Models trained from data containing

conversations between many interlocutors are able to perform some common-sense reasoning, it has been suggested that attempting to model a single speaker ("persona") may yield qualitative improvements in perplexity and BLEU scores (Li et. al, 2016); this is consistent with anthropological approaches to the relationship between human intelligence and sociality/context. In 2018, several important breakthroughs in NLP were achieved using universal transformer models trained in an unsupervised manner on large quantity of text from the web. Examples include Google's BERT and OpenAI's generative pre-trained transformers (GPT and GPT-2).

Team Roles and Responsibilities

Team Member	Responsibilities
Alexander J. Orona	Modeling, APIs and App
Martin Malasky	AWS Infrastructure

Data Acquisition

Data Sources and Data Collection

The main dataset was a csv file of dialog from the character "Rachel" from friends. This data is was sequential data, which we filtered for occurrences of dialog containing Rachel. We organized this into input-output pairs, where input represents something said to Rachel and output represents something she said in return. The transformed baseline dataset was composed of approximately 7,300 rows and was approximately 5MB in size. This manageable dataset was augmented in various ways over the project. Given the manageability of the dataset, pandas dataframes were sufficient to handle all data in memory.

Production data generated by users are also very small. The rest API ingests JSON files of dialogue containing at most a thousand words. History is stored locally in the user's browser, so there is no necessary database.

The prepared dataset used for the TransferTransfo model organizes inputs in a more complicated fashion. The pre-tokenized input includes the 'persona' (or context) of the utterance and a history of the dialogue, including the most recent utterance. This is rendered as a JSON file during the training process. Tokens mark speakers. It is 5.6MB in size.

INITIAL BASELINE DATASET SUMMARY

	conversation	input_speaker	gender	input	output_speaker	output
0	1	MONICA	F	Rachel?!	RACHEL	Oh God Monica hi! Thank God! I just went to yo...
1	1	MONICA	F	De-caff. Okay, everybody, this is Rachel, anot...	RACHEL	Hi, sure!
2	1	MONICA	F	So you wanna tell us now, or are we waiting fo...	RACHEL	Oh God... well, it started about a half hour b...
3	1	MONICA	F	Who wasn't invited to the wedding.	RACHEL	Ooh, I was kinda hoping that wouldn't be an is...
4	2	ROSS	M	I'll have whatever Christine is having.	RACHEL	Daddy, I just... I can't marry him! I'm sorry....

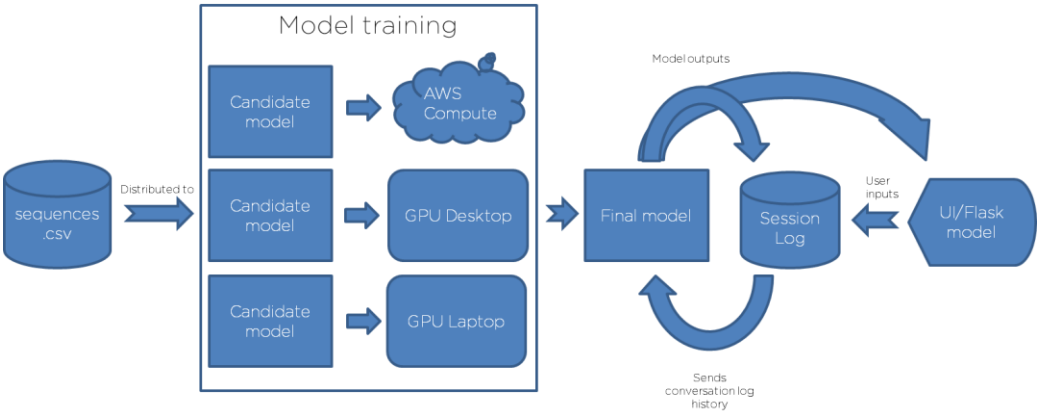
SAMPLE OF INPUT TO TRANSFER-TRANSFO MODEL

<bos>my name is rachel green, but you already knew that. i am not dating ross. Monica and phoebe are my closest friends. <MONICA>rachel?!<RACHEL>Oh God Monica hi! Thank God! I just went to your building and you weren't there and then this guy with a big hammer said you might be here and you are, you are!<eos>

Data Pipelines

Our training architecture utilized both cloud and local machines. Since training and inference require GPUs, all of these machines had varying capabilities. Initial training setup and trial runs took place on local machiness. Once models were debugged, longer runs were pushed to our main AWS instance (transfer-1). The final TransferTranfo model was an universal transformer, which was memory intensive. This required a Tesla v100 16gb GPU. To conserve resources, we used the smallest p3 instance: p3.2xlarge.

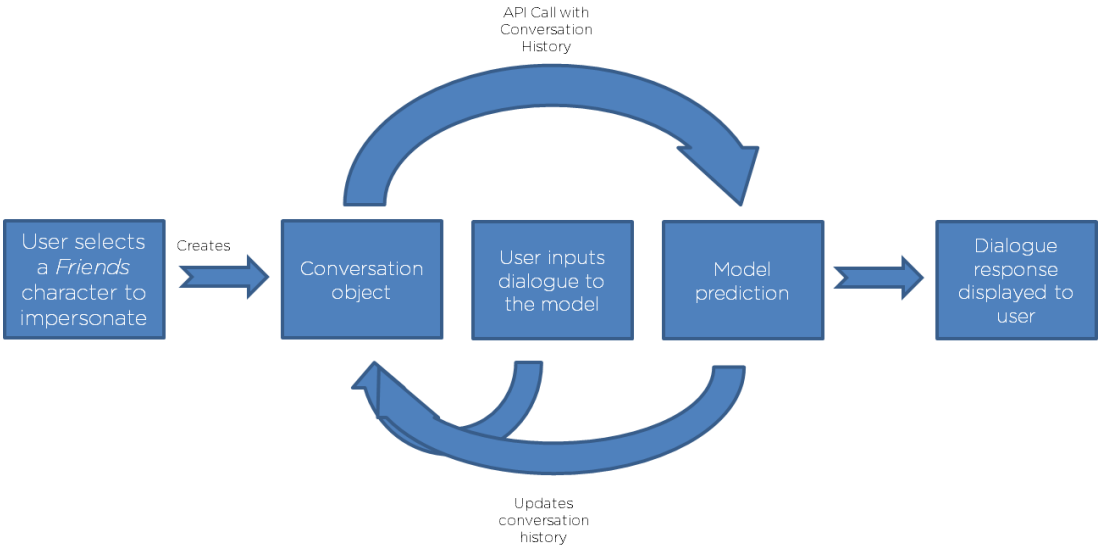
TRAINING DATA PIPELINE AND INFRASTRUCTURE



Production infrastructure is comprised of two Flask applications. The first is a rest API that ingests a dialogue history and impersonation value (“Monica”, “Ross”, etc.) as a JSON file. Essentially, it exposes the model. This API runs in the inference environment on an AWS instance. The second application is the user front-end chatbot. This can run on any machine in our flask environment. The chatbot app collects user dialogue,

stores the dialogue history and send it to the model API for prediction. It then posts the results to the user interface.

PRODUCTION DATA PIPELINE AND INFRASTRUCTURE



Data Preparation

An essential finding of exploratory data analysis was the detection of a major problem in the dataset. We initially used the sequences.csv data for this deeper dive. However, after an initial word cloud visualization of additional speakers, we noticed that all of the word clouds showed no obvious patterns of difference. After further manual investigation of the several of the rows, we discovered that the dataset contained some error for the speaker identifiers. Following this discovery, it was necessary to reconstruct the dataset from a scrapped source. After some simple data cleaning in Python and manual verification of the veracity of the dataset using a repository of Friends transcripts online, we created a new csv containing all sequential lines between all characters. We filtered this larger dataset for just those lines Rachel responded to. This corrected the problem.

After moving to the TranserTransfo architecture, significant data preparation and dataset curation had to take place. It was necessary to provide a new context input for each row. This was achieved by focusing a few aspects of Rachel's persona:

- Her basic identity (name)
- Her relationship status with Ross
- The identities of her best friends (Phoebe and Monica)

This was done intentionally, as only the relationship status with Ross changed – the other two were constant – enabling future exploration of whether this makes a difference in responses to/about the speaker.

Furthermore, the TransferTransfo architecture required the generation of distractors and positional embeddings. This process was very akin to feature generation, but highly qualitative. Exploring this space sufficiently remains an open question. The initial changes were accomplished via data processing notebooks.

INPUT FORMATTING FOR TRANSFER-TRANSFO

Persona (Context) + Target	Persona (Context) + Distractor	Positional Encoding
<bos>my name is rachel, but you already knew that. i am not dating ross. Monica and phoebe are my closest friends. <MONICA>rachel?! <RACHEL> Oh God Monica hi! Thank God! I just went to your building and you weren't there and then this guy with a big hammer said you might be here and you are, you are!<eos>	<bos>my name is rachel, but you already knew that. i am not dating ross. Monica and phoebe are my closest friends. <MONICA>rachel?! <RACHEL> of urban (its members may belong first in <eos>	[-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1]

Analysis Methods

Preliminary analysis was conducted using LDA and WordClouds. This exploratory data analysis provided a number of insights:

- Rachel's language is not like her interlocutors
- Her language changes based on who she is talking to
- Idiosyncrasies are not 'noise' and caution should be used in eliminating 'stopwords'
- The dataset is filled with scene-based dialogue
- A good model will need to take account of the fact that a chatbot context is different from a drama. Only in the very last part of the project was it possible to come to terms with what this means.

As a result of these insights, we did not stem or remove stopwords and we paid close attention to the correspondence of model language to Rachel's syntax and style during the training process.

EXPLORATORY DATA ANALYSIS WORDCLOUDS

Rachel to Monica

Rachel to Ross

Phoebe

Rachel to Chandler

Rachel to Joey

Modeling

A very large portion of time in this project was spent on finding a good model for Rachel. Since the preliminary literature review predicted various recurrent architectures as the main method for building dialogue agents, much of the effort was spent on recurrent architecture searches. After failing to converge on validation for hand-crafted models, a 17-model search was conducted using the Ludwig library. None of these models was able to obtain an ADEM score better than 2.8.

FAILURE TO CONVERGE ON VALIDATION DATASET (HAND-CRAFTED ARCHITECTURE)

Run Statistics

Training Time: 20h 31m

Epochs: 109

Error

Starting

Training Loss: 0.4203

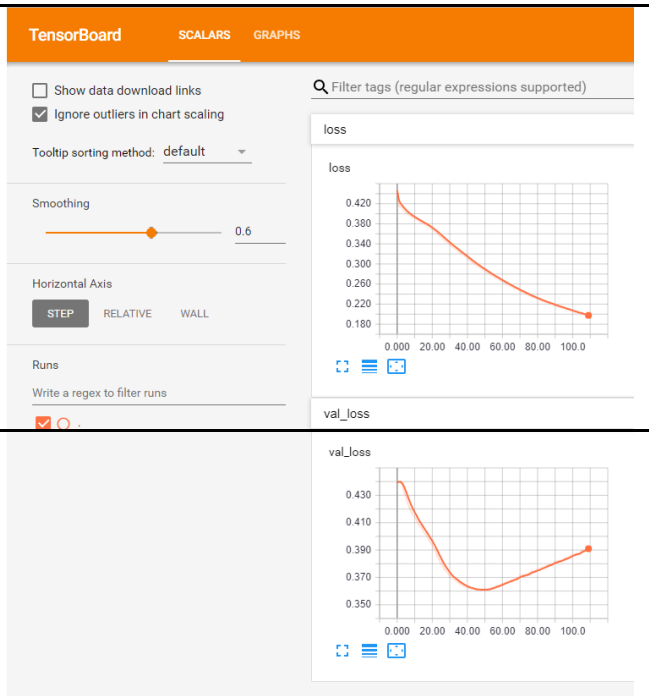
Validation Loss: 0.4393

Final

Training Loss: 0.1976

Validation Loss: 0.3910

ADEM Score: 2.78



Example Outputs

Q: oh , i'm sorry .
A: oh , you guys !

Q: isn't he great ?
A: i don't know . do-do you know my god , i have a baby .

Q: how are you doing?
A: (Rachel): i don't know . i mean , i know , chris o'donnel , john f . kennedy , jr .

LUDWIG ARCHITECTURE SEARCH

TIER 1 - MODEL PERFORMANCE

Name	Directionality	Embedding	Search	Overfit (validation loss – train loss)	Validation Loss (compared to best)
t1_gbb	bidirectional	glove	beam	0.37152	0.08322
t1_gbu	unidirectional	glove	beam	0.9818	0.0833
t1_ggb	bidirectional	glove	greedy	0.9669	0
t1_ggu	unidirectional	glove	greedy	0.8786	0.0644
t1_nbb	bidirectional	none	beam	1.0033	0.0808
t1_nbu	unidirectional	none	beam	1.1297	0.0364
t1_ngb	bidirectional	none	greedy	0.8221	0.068
t1_ngu	unidirectional	none	greedy	0.8907	0.1666

TIER 1 - ARCHITECTURAL ELEMENT PERFORMANCE

Parameter	Type	Average Model Overfit (validation loss – train loss)	Average Model Validation Loss (compared to best)
bidirectional	Directionality	0.790955	0.058005
unidirectional	Directionality	0.9702	0.087675
glove	Embedding	0.799705	0.05773
none	Embedding	0.96145	0.08795
beam	Search	0.87158	0.07093
greedy	Search	0.889575	0.07475

TIER 1 - SUMMARY

Best model by performance: bidirectional, glove, greedy
Best predicted architectural elements: bidirectional, glove, beam
Selected elements: bidirectional, glove, beam

TIER 2 - MODEL PERFORMANCE

Name	Encoder Type	Attention	Overfit (validation loss – train loss)	Validation Loss (compared to best)

t2_pb	parallel cnn	bahdanau	0.8837	0
t2_pl	parallel cnn	luong	0.8677	0.0064
t2_sb	stacked cnn	bahdanau	1.1258	0.1034
t2_sl	stacked cnn	luong	0.7234	0.0224
t2_2b	stacked parallel cnn	bahdanau	0.7584	0.0674
t2_2l	stacked parallel cnn	luong	0.771	0.0677
t1_gbb	rnn	bahdanau	0.37152	0.07442
t2_rl	rnn	luong	1.1258	0.1034
t2_cb	cnn rnn	bahdanau	1.1889	0.3381
t2_cl	cnn rnn	luong	1.0649	0.1735

TIER 2 - ARCHITECTURAL ELEMENT PERFORMANCE

Parameter	Type	Average Model Overfit (validation loss – train loss)	Average Model Validation Loss (compared to best)
cnn rnn	Encoder Type	1.1269	0.2558
parallel cnn	Encoder Type	0.8757	0.0032
rnn	Encoder Type	0.74866	0.08891
stacked cnn	Encoder Type	0.9246	0.0629
stacked parallel cnn	Encoder Type	0.7647	0.06755
bahdanau	Attention	0.865664	0.116664
luong	Attention	0.91056	0.07468

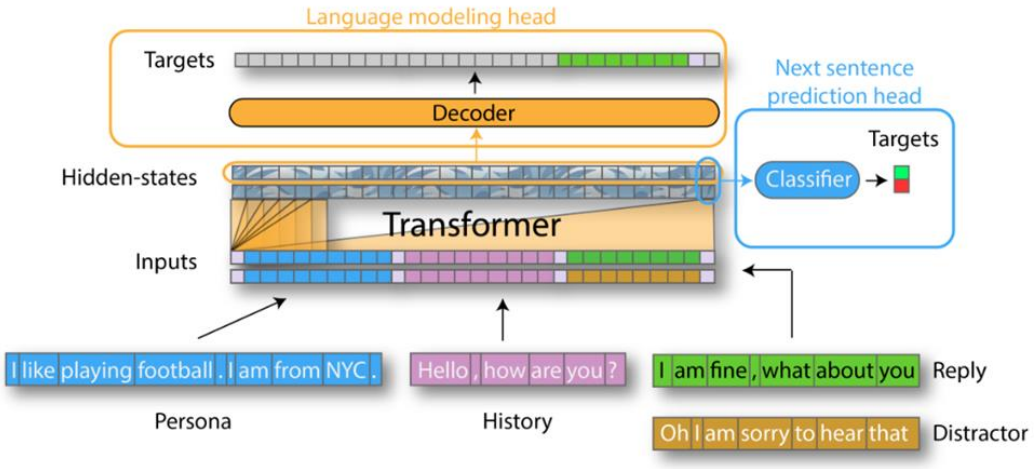
TIER 2 - SUMMARY

Best model by performance: parallel cnn, bahdanau
Best predicted architectural elements: parallel cnn, luong
Selected elements: parallel cnn, luong

TRANSFER-TRANSFO

Transfer learning using large pre-trained transformers emerged as a major topic in late 2018 and early 2019. Following the failure of RNNs to converge on validation loss, this approach was taken. This scheme allowed utilization of a pretrained general language model released by OpenAI called generative pretrained transformer ('gpt'). In the recent ConvAI2 Challenge hosted by NIPS (NeurIPS), significant improvements on the state-of-the-art were achieved by Thomas Wolf and his team at Hugging Face using this methodology. After Thomas Wolf published his methodology in February and open sourced his ConvAI2 code in May, this technique was taken. Wolf's libraries were modified to fit the Rachel problem, resulting in this training architecture:

FINAL MODEL TRAINING ARCHITECTURE (FOLLOWING WOLF ET. AL 2019)

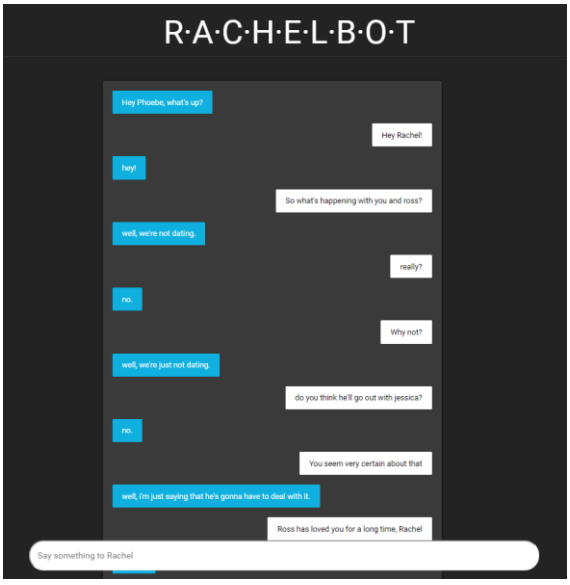


Preliminary results indicated an ADEM score of 3.2 to 3.5 depending on variations in context (persona) inputs.

Findings and Reporting

Chatbots are easy to present, but easy to misinterpret. The main visual output of this project is a chatbot which can be interacted with and used. That enables us to meet our goal of creating an engaging Rachel experience. In this report, it was crucial that the timeline and milestones be documented: a chatbot is deceptively simple, and so the extent of the work and the research findings cannot be adequately condensed into a chat screen.

THE RACHELBOT



Solution Architecture, Performance and Evaluation

The main measure of performance was a model of human chatbot scoring proposed by Lowe et. al (2017). Lowe et. al proposed to train a model to ingest 1) an utterance to a chatbot, 2) a chatbot response, and 3) the actual label in order to predict a human score. We utilized a pretrained Theanos model implementing Lowe et. al's methodology to generate scores (very slowly). The ADEM scorer took several hours to days to score an entire dataset, so we used this sparingly. None of our RNN variations scored better than 2.8. The first model build of the TransferTransfo framework received a 3.2, with subsequent modifications reaching as high as 3.5.

Management of our budget on AWS was done by using the smallest AWS instance in development and small test runs, upgrading only when we were certain that a particular code base for training had been implemented correctly.

Conclusions

With general language modeling, the core challenge to improvement is understanding context. Recurrent neural network architectures may perform well in certain natural language tasks, such as machine translation, but they are unlikely to generate the memory analysis required for meaningful conversational agents. As with other researcher, general recurrent neural networks applied to this problem generated inconsistent outputs and the lack of a consistent personality (Li and Jurafsky, 2016), struggled to deal with the context of the dialogue, tended to produce consensual and generic response which were not engaging (Li, Monroe, and Jurafsky, 2016) and could not capture Rachel. In contrast, over-trained recurrent neural networks did capture Rachel – a crazy, unpredictable version of the character.

At the onset, we knew that a truly interactive agent would be difficult to capture with the state of the art. It could still be as difficult as developing a general artificial intelligence. However, after this project it is clear that context management with universal transformers could enable very good conversational models to appear. The final models were able to exceed ADEM scores substantially (2.8 for RNNs, 3.2+ for transformer models). There is still much to be done to improve these models. However, this project suggests that a wide range of enterprise applications may be on the horizon if research can improve on the latest paradigm shift in conversational agent construction.

Resources

Chen, Hongshen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. ACM SIGKDD Explorations Newsletter 19(2): 25-35.

Gatt, Albert, Emiel Krahmer. 2018. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. Journal of Artificial Intelligence Research 61(1): 65-170.

He, He, Anusha Balakrishnan, Mihail Eric, Percy Liang. 2017. Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).

Li, Jiwei, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, Bill Dolan. 2016. A persona-based neural conversation model. Retrieved from arXiv.org on 20 October 2018 at https://arxiv.org/abs/1603.06155.

Li, Jiwei, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. Retrieved from arXiv.org on 20 October 2018 at https://arxiv.org/abs/1606.01541.

Li, Jiwei, Will Monroe, Tianlin Shi, Sebastien Jean, Alan Ritter, Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. Retrieved from arXiv.org on 20 October 2018 at https://arxiv.org/abs/1701.06547.

Liu, Bing, Ian Lane. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. Retrieved from arXiv.org on 20 October 2018 at https://arxiv.org/abs/1709.06136.

Liu, Chia-Wei, Ryan Lowe, Iulian V. Serban, Michael Noseworthy, Laurent Charlin, Joelle Pineau. 2017. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.

Lowe, Ryan, Michael Noseworthy, Iulian V. Serban, Nicolas Angelard-Gontier, Yoshua Bengio, Joelle Pineau. 2017. Towards an automatic Turing test: learning to evaluate dialogue responses. Retrieved from arXiv.org on 20 October 2018 at https://arxiv.org/abs/1708.07149.

Luan, Yi, Yangfeng Ji, Mari Ostendorf. 2016. LSTM based conversation models. Retrieved from arXiv.org on 20 October 2018 at https://arxiv.org/abs/1603.09457.

Trask, Andrew, Felix Hill, Scott Reed, Jack Rae, Chris Dyer, Phil Blunsom. 2018. Neural arithmetic logic units. Retrieved from arXiv.org on 20 October 2018 at https://arxiv.org/abs/1808.00508.

Vinyals, Oriol, Quoc Le. 2015. A neural conversational model. Retrieved from

arXiv.org on 20 October 2018 at https://arxiv.org/abs/1506.05869.

Wen, Tsung-Hsien, Yishu Miao, Phil Blunsom, Steve Young. 2017. Latent intention dialogue models. Retrieved from arXiv.org on 20 October 2018 at https://arxiv.org/abs/1705.10229.

Williams, Jason D., Kavosh Asadi, Geoffrey Zweig. 2017. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).

Young, Tom, Devamanyu Hazarika, Soujanya Poria, Erik Cambria. 2018. Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine.

Young, Tom, Erik Cambria, Iti Chaturvedi, Minlie Huang, Hao Zhou, Subham Biswas. 2018. Augmenting end-to-end dialog systems with commonsense knowledge. The Thirty-Second AAAI Conference on Artificial Intelligence.

Zhou, Hao, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. Retrieved from arXiv.org on 20 October 2018 at https://arxiv.org/abs/1704.01074.

Appendix A

DSE MAS Knowledge Applied to Project

- DSE 200 (Introduction to Python): utilized extensively in dataset manipulation
- DSE 210 (Advanced Statistics): evaluation of metrics of performance
- DSE 220 (Machine Learning: supervised machine learning, especially neural networks
- DSE 241 (Visualization): extensive application of visual presentation principles in exploratory data analysis and in user interface design of chatbot
- Capstone 260A and 260B: critical project management and documentation skills, as well as processes for complex projects from inception to delivery