Multi-Task Learning for Speaker-Role Adaptation in Neural Conversation Models

Yi Luan[†*] **Chris Brockett**[‡] **Bill Dolan**[‡] **Jianfeng Gao**[‡] **Michel Galley**[‡]
[†]Department of Electrical Engineering, University of Washington
[‡]Microsoft Research
luanyi@uw.edu, {chrisbkt,billdol,jfgao,mgalley}@microsoft.com

Abstract

Building a persona-based conversation agent is challenging owing to the lack of large amounts of speaker-specific conversation data for model training. This paper addresses the problem by proposing a multi-task learning approach to training neural conversation models that leverages both conversation data across speakers and other types of data pertaining to the speaker and speaker roles to be modeled. Experiments show that our approach leads to significant improvements over baseline model quality, generating responses that capture more precisely speakers' traits and speaking styles. The model offers the benefits of being algorithmically simple and easy to implement, and not relying on large quantities of data representing specific individual speakers.

1 Introduction

Conversational engines are key components of intelligent "personal assistants" such as Apple's Siri and Amazon's Alexa. These assistants can perform simple tasks, answer questions, provide recommendations, and even engage in chit-chats (De Mori et al., 2008; Chen et al., 2015, 2016). The emergence of these agents has been paralleled by burgeoning interest in training natural-sounding dialog systems from conversational exchanges between humans (Ritter et al., 2011; Sordoni et al., 2015; Luan et al., 2014, 2015; Vinyals and Le, 2015). A major challenge for data-driven systems is how to generate output that corresponds to specific traits that the agent needs to adopt, as they tend to generate "consensus" responses that are often commonplace and uninteresting (Li et al., 2016a; Shao et al., 2017).

User input: *I am getting a loop back to login page.*
Baseline model: Ah, ok. Thanks for the info.
Our model: I'm sorry to hear that. Have you tried clearing your cache and cookies?

Figure 1: Existing neural conversational models (baseline) tend to produce generic responses. The system presented in this paper better represents the speaker role (support person), domain of expertise (technical), and speaking style (courteous).

This is illustrated in Fig. 1, where the output of a standard Sequence-to-Sequence conversation model is contrasted with that of the best system presented in this work. The baseline system generates a desultory answer that offers no useful information and is unlikely to inspire user confidence. The output of the second system, however, strongly reflects the agent's role in providing technical support. It not only evidences domain knowledge, but also manifests the professional politeness associated with a speaker in that role.

The challenge for neural conversation systems, then, is that an agent needs to exhibit identifiable role-specific characteristics (a 'persona'). In practice, however, the conversational data needed to train such systems may be scarce or unavailable in many domains. This may make it difficult to train a system represent a doctor or nurse, or a travel agent. Meanwhile, appropriate non-conversational data (e.g., blog and micro-blog posts, diaries, and email) are often abundant and may contain much richer information about the characteristics of a speaker, including expressive style and the role they play. Yet such data is difficult to exploit directly, since, not being in conversational format, it does not mesh easily with existing source-target conversational models.

* This work was performed at Microsoft.

In this paper we address the joint problems of blandness and data scarcity with multi-task learning (Caruana, 1998; Liu et al., 2015; Luan et al., 2016a). This is a technique that has seen success in machine translation, where large monolingual data sets have been used to improve translation models (Sennrich et al., 2016). The intuition is that if two tasks are related, then joint training and parameter sharing can enable one task to benefit the other. In our case, this sharing is between two models: On one hand, a standard Sequence-to-Sequence conversational models is trained to predict the current response given the previous context. On the other hand, using the non-conversational data, we introduce an autoencoder multi-task learning strategy that predicts the response given the same sequence, but with the target parameters tied with the general conversational model.

Our experiments with 4M conversation triples show that multi-task adaptation is effective in that the generated responses capture speaker-role characteristics more precisely than the baseline. Experiments on a corpus of Twitter conversations demonstrate that multi-task learning can boost performance up to 46.2% in BLEU score and 23.0% in perplexity, with a commensurate consistency gains in human evaluation.

2 Related Work

2.1 Conversational Models

In contrast with much earlier work in dialog, our approach to conversation is wholly data-driven and end-to-end. In this respect, it follows a line of investigation begun by (Ritter et al., 2011), who present a statistical machine translation based conversation system. End-to-end conversation models have been explored within the framework of neural networks (Sordoni et al., 2015; Vinyals and Le, 2015; Li et al., 2016a,b; Luan et al., 2017). The flexibility of these Sequence-to-Sequence (SEQ2SEQ) encoder-decoder neural models opens the possibility of integrating different kinds of information beyond the single previous turn of the conversation. For example, (Sordoni et al., 2015) integrate additional contextual information via feed-forward neural networks. (Li et al., 2016a) use Maximum Mutual Information (MMI) as the objective function in order to produce more diverse and interesting responses. (Mei et al., 2017) introduce an attention mechanism into an encoder-decoder network for a conversation model.

(Wen et al., 2015) introduced a Dialog-Act component into the LSTM cell to guide generated content. (Luan et al., 2016b) use a multiplicative matrix on word embeddings to bias the word distribution of different speaker roles. That work, however, assumes only two roles (questioner and answerer) and is less generalizable than the model proposed here.

Most relevant to the present work, (Li et al., 2016b) propose employing speaker embeddings to encode persona information and allow conversation data of similar users on social media to be shared for model training. That work focused on individuals, rather than classes of people. The approach, moreover, is crucially dependent on the availability of large-scale conversational corpora that closely match the persona being modeled—data that, as we have already observed, may not be readily available in many domains. In this work, we circumvent these limitations by bringing non-conversation corpora (analogous to the use of monolingual data in machine translation) to bear on a general model of conversation. Doing so allows us to benefit in terms of representing both the role of the agent and domain content.

2.2 Multi-Task Learning

Multi-task learning has been successfully used to improve performance in various tasks, including machine translation (Sennrich et al., 2016) and image captioning (Luong et al., 2016). (Sennrich et al., 2016) report methods of exploiting monolingual data—usually available in much larger quantities—to improve the performance of machine translation, including multi-task learning of a language model for the decoder. Autoencoders are widely used to initialize neural networks (Dai and Le, 2015). (Luong et al., 2016) show that an autoencoder of monolingual data can help improve the performance of bilingual machine translation in the form of multi-task learning. In our models, we share the decoder parameters of a SEQ2SEQ model and autoencoder to incorporate textual information through multi-task learning.

3 Background

3.1 Task definition

The task of response generation is to generate a response given a context. In this paper, following (Sordoni et al., 2015), each data sample is

represented as a *(context,message,response)* triple, where context is the response of the previous turn, and the message is the input string of the current turn. The response, then, is the sequence to be predicted given these two strings of information. In addition to the triple, large-scale non-conversational data from the responder is provided as side information.

3.2 Sequence-to-Sequence Conversational Models

Given a sequence of inputs $X = \{x_1, x_2, \ldots, x_{n_X}\}$ and the corresponding output $Y = \{y_1, y_2, \ldots, y_{n_Y}\}$, Sequence-to-Sequence (SEQ2SEQ) models use a Long Short-Term Memories (LSTM) (Hochreiter and Schmidhuber, 1997) to encode the input sequence, taking the last hidden state of encoder h_{n_X} to represent output sequence. The decoder is initialized by h_{n_X}, and predict output y_t given h_{n_X} and y_{t-1}.

Our input is context followed by message, delimited by an *EOS* token. The LSTM cell includes an input gate, a memory gate and an output gate, respectively denoted as i_t, f_t and o_t.

3.3 Persona-based conversational model

The persona-based conversational model is a variant of standard SEQ2SEQ models, with user information encoded at decoder. As in standard SEQ2SEQ models, the persona-based conversational model presented in (Li et al., 2016b) first encodes the source message into a vector representation using the source LSTM. Then, for each element in the target side, hidden units are obtained by combining the representation produced by the target LSTM at the previous time step h_{t-1}, the word representations e_t at the current time step, and the embedding s_i for user i.

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ l_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \cdot \begin{bmatrix} h_{t-1} \\ e_t \\ s_i \end{bmatrix} \quad (1)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot l_t \quad (2)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (3)$$

where $W \in \mathbb{R}^{4K \times 3K}$. This model assigns one K dimensional vector representation to each of the speakers in the corpus. It thus relies on the availability of sufficient conversational training data of each speaker to learn meaningful speaker embeddings. Since this type of data is usually hard to obtain in real application scenarios, we need a method that can leverage easier-to-obtain non-conversational personal data in order to incorporate richer personal information into conversational models.

4 A Multi-task Learning Approach

Given the limitations of previous methods, we propose the following multi-task learning approach in order to simultaneously leverage conversational data across many users on the one hand, and personal but non-conversation data (written text) of a specific user on the other. We define the following two tasks:

- A **SEQ2SEQ** task that learns conversational models described in Section 3 using conversation data of a large general population of speakers.
- An **AUTOENCODER** task that utilizes large volumes of non-conversational personal data from target speakers.

AUTOENCODER: An AUTOENCODER is an unsupervised method of obtaining sequence embeddings based on the SEQ2SEQ framework. Like a SEQ2SEQ model, it comprises encoding and decoding components built by an LSTM sequential model as in Section 3.2. Instead of mapping source to target as in a SEQ2SEQ model, the AUTOENCODER predicts the input sequence itself.

Parameter sharing: Given the same context, we want to generate a response that can mimic a particular target speaker. Therefore, we share only the decoder parameters of SEQ2SEQ and AUTOENCODER while performing multi-task learning, so that the language model for generation can be adapted to the target-speaker . Since the context is not constrained and can be from any speaker, the encoder parameters are not tied and are learned separately by each task. (See Fig. 2.)

Training Procedure The training procedure is shown in Fig. 3. In each iteration, the gradient of each task is calculated according to the task-specific objective. The training process finishes when perplexity performance converges in dev set and the best model is selected according to SEQ2SEQ perplexity performance.

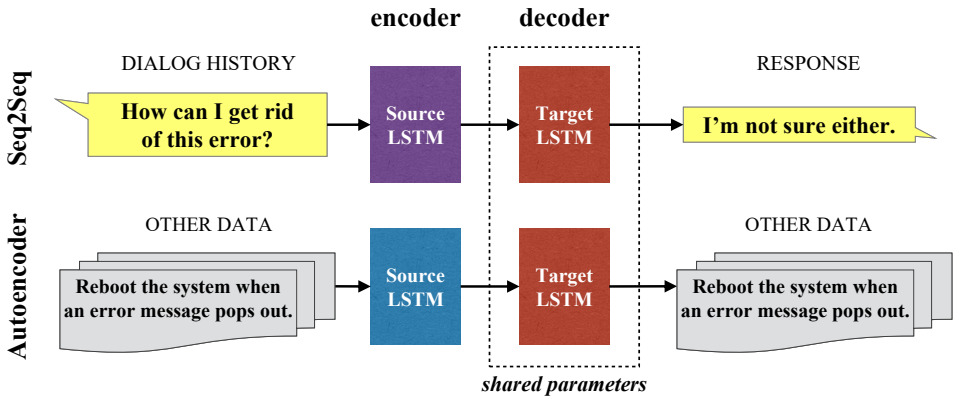

Figure 2: Framework of Multi-task learning. The parameters of decoder are shared across the two tasks.

Training procedure of Multi-task learning:

1. Randomly initialize SEQ2SEQ and AUTOENCODER encoder parameters.
2. Train SEQ2SEQ model until dev set performance converges in perplexity.
3. **While** not dev set performance converged in perplexity **do**:
   (a) Randomly pick a batch of samples from general conversational data.
   (b) Compute loss and gradient for SEQ2SEQ task and update parameters.
   (c) Randomly pick a batch of samples from non-conversational data of the target user.
   (d) Compute loss and gradient for AUTOENCODER task and update parameters.
4. Choose the best model based on SEQ2SEQ perplexity performance on dev set.

Figure 3: Training Procedure

5 Single v.s. Multiple speaker Settings

Two variants of SEQ2SEQ task are explored:

- **MTASK-S** Personalized response generation for a **single** user, which uses the basic SEQ2SEQ conversational model as described in Section 3.2.
- **MTASK-M** Response generation for **multiple** users, which uses the persona-based SEQ2SEQ model described in Section 3.3.

MTASK-S: We train a personalized conversational model for one speaker at a time. For each target user, we need to perform separate multi-task training which results in N models for N users. This is inefficient in both memory and computational cost.

MTASK-M: In order to address the memory and computation issue of MTASK-S, we introduce user embeddings to SEQ2SEQ model as in Eq. 1. We first train a persona-based conversational model using conversational data for a general population of speakers. This model differs MTASK-S in that it introduces two parameter matrices into the decoder: a speaker embedding s_i and its corresponding weight matrix that can decouple speaker dependent information from general language information. In the multi-task stage, since the target users have never appeared in the training data, we randomly initialize the user embeddings for those users and follow the training procedure as in Figure 3.[1] The embedding of the unseen user is updated by AUTOENCODER training together with the decoder LSTM parameters.

6 Experimental Setup

6.1 Datasets

As training data, we use a collection of 3-turn conversations extracted from the Twitter FireHose. The dataset covers the six-month period beginning January 1, 2012, and was limited to conversations where the responders had engaged in at least 60 3-turn conversational Twitter interactions during the period. In other words, these are people who reasonably frequently engaged in conversation, and might be experienced "conversationalists."

We selected the top 7k Twitter users who had most conversational data from that period (at least 480 turns, average: 571). This yielded a total of approximately 4M conversational interactions. In addition to these 7k general Twitter users, we also selected the 20 most frequent users, employing all of their conversation data for development and test. Twitter users typically have many more single posts than posts that interact with other people.

[1] The model can also be learned without pre-training (omitting step 2), but we found that pre-training usually helps.

We therefore treat single posts as non-conversation data. All single posts of the 20 top users (at least 9k per user, average 10.3k) were extracted for multi-task learning. The 20 users were of diverse backgrounds, including technical support personnel, novelists, and sports fans.

6.2 Evaluation

As in previous work (Sordoni et al., 2015), we use BLEU and human evaluation for evaluation. BLEU (Papineni et al., 2002) has been shown to correlate fairly well with human judgment at a document- and corpus-level, including on the response generation task.[2] We also report perplexity as an indicator of model capability.

We additionally report degree of diversity by calculating the number of distinct unigrams and bigrams in generated responses. The value is scaled by total number of generated tokens to avoid favoring long sentences (shown as distinct-1 and distinct-2). Finally, we present a human evaluation that validates our main findings.

6.3 Baseline

Our baseline is our implementation of the LSTM-MMI of (Li et al., 2016a). The MMI algorithm reduced the blandness of SEQ2SEQ models by scoring the generated N-best list with a function that linearly combines a length penalty and the log likelihood of source given target:

$$\log p(R|M,v) + \lambda \log p(M|R) + \gamma|R| \quad (4)$$

where $p(R|M,v)$ is the probability of the generated response given message M and the respondents user ID. $|R|$ is the length of the target and γ is the associated penalty weight. We use MERT (Och, 2003) to optimize γ and λ on BLEU using N-best lists of response candidates generated from the development set. To compute $p(M|R)$, we train an inverse SEQ2SEQ model by swapping messages and responses. The reverse SEQ2SEQ models $p(M|R)$ is trained with no user information considered.

6.4 Training and Decoding

We trained two-layer SEQ2SEQ models on the Twitter corpus, using the following settings:

[2](Liu et al., 2016) suggest that BLEU doesn't correlate well with human judgment at the sentence level. Other work, however, has shown that correlation increases substantially with larger units of analysis (e.g., document or corpus) (Galley et al., 2015; Przybocki et al., 2009).

	Baseline	MTASK-S	MTASK-M
Perplexity (dev)	56.33	**32.27** (-42.7%)	44.96 (-20.2%)
Perplexity (test)	61.17	**39.83** (-34.9%)	43.21 (-29.4%)

Table 1: Perplexity for standard SEQ2SEQ and the user model on the Twitter Persona dev set.

	Baseline	MTASK-S	MTASK-M
BLEU (dev)	1.32	1.76 (+33.3%)	**2.52** (+90.1%)
BLEU (test)	1.31	1.69 (+29.0%)	**2.25** (+71.7%)
distinct-1	1.69%	2.43%	**2.44%**
distinct-2	6.53%	**10.2%**	9.79%

Table 2: Performance on the Twitter dataset of 2-layer SEQ2SEQ models and MMI models. Distinct-1 and distinct-2 are respectively the number of distinct unigrams and bigrams divided by total number of generated words.

- 2 layer LSTM models with 500 hidden cells for each layer.
- Batch size is set to 128.
- Optimization method is Adam (Kingma and Ba, 2015).
- Parameters for SEQ2SEQ models are initialized by sampling from uniform distribution $[-0.1, 0.1]$.
- Vocabulary size is limited to 50k.
- Parameters are tuned based on perplexity.

For decoding, the N-best lists are generated with beam size $B = 50$. The maximum length of the generated candidates was set at 20 tokens. At each time step, we first examine all $B \times B$ possible next-word candidates, and add all hypotheses ending with an EOS token to the N-best list. We then preserve the top-B unfinished hypotheses and move to the next word position. We then use LSTM-MMI to rerank the N-best list and use the 1-best result of the re-ranked list in all evaluation.

7 Experimental Results

The perplexity and BLEU score results for three models are shown in Tables 1 and 2. Compared with the baseline model LSTM-MMI, we obtain a 34.9% decrease in perplexity for the MTASK-S model and a 29.4% decrease in perplexity for the MTASK-M model. Significant gains are obtained in BLEU score as well: MTASK-S gains

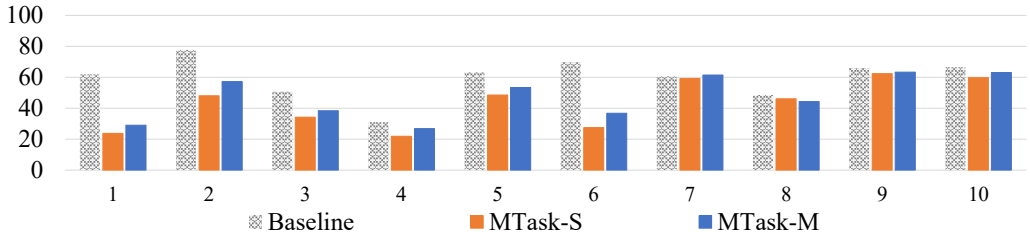

Figure 4: Perplexity scores for the top 10 users with most (non-conversational) training data. Users with obvious speaking styles or stronger user role characteristics (e.g., 1, 2, and 6) show much greater perplexity reduction than the other ones.

29.0% relative increase compared with the baseline and MTASK-M gains 71.7%. MTASK-S performance is better than MTASK-M in perplexity, but worse on BLEU score. Apart from the fact that BLEU does not necessarily correlate with perplexity, this result also indicates that MTASK-S has more parameters (each user has a unique model for MTASK-S) but tends to overfit on development set perplexity. Another possible reason that MTASK-M performs better than MTASK-S is the introduction of user embeddings. The persona-based conversational model can decouple the personalized information from general language patterns and can therefore encode user characteristic better. We further report degree of diversity by calculating the number of distinct unigrams (distinct-1) and bigrams (distinct-2) in generated responses as in Table 2. To avoid biasing toward longer sentences, this value is scaled by the total number of tokens generated. Both MTASK-S and MTASK-M models perform better than baseline in terms of distinct-1 and distinct-2, which we interpret to mean that our approach can help the system generate responses that are more diverse yet better approximate the targeted speaker or speaker type.

Fig. 4 shows the perplexities for the 10 individual users most represented in the non-conversational training data. Our multi-task approaches consistently outperform baseline on perplexity. However, the performance between individual target users can vary substantially.[3]

After inspecting dev set outputs, we observe that users with obvious speaking styles or stronger user role characteristics show much greater gain than the others. For example, User 1 is a technical support worker who answers web questions for Twitter users, while User 2 always expresses strong feelings and uses exclamation marks frequently. Conversely, tweets from users that did not show significant gain appear to be more about daily life and chitchat, with no strong role characteristics (e.g., Users 3 and 4). We present example outputs for User 1 and 2 in Section 8.

7.1 Human Evaluation

Human evaluation of the outputs was performed using crowdsourcing.[4] Evaluation took the form of a preference test in which judges were presented with a random sample of 5 tweets written by the targeted user as example texts, and asked which system output appeared most likely to have been produced by the same person. A 5-point scale that permitted ties was used, and system pairs were presented in random order. A short input message (the input that was used to generate the outputs) was also provided. We used 7 judges for each comparison; those judges whose variances differed by more than two standard deviations from the mean variance were discarded. Table 3 shows the results of pairwise evaluation, along with 95% confidence intervals of the means. MTASK-S and MTASK-M both perform better on average than LSTM-MMI, consistent with the BLEU results. MTASK-M's gain over the LSTM-MMI baseline is significant at the level of $\alpha = 0.05$ ($p = 0.026$), indicating that judges were better able to associate the output of that model with the target author.

In Table 3 the strength of the trends is obscured by averaging. We therefore converted the scores for each output into the ratio of judges who selected that system for each output (Figs. 5 and 6). To read the charts, bin 7 on the left represents

[3]We do not report BLEU scores for individual users, as the dev and test set for each specific user tends to be small (less than 500 samples) and BLEU is known to be unreliable when evaluated on small datasets (Graham et al., 2015; Liu et al., 2016).

[4]Two outputs were removed from the datasets owing to offensive content in the examples.

	Baseline	System
MTASK-S	0.491 ±0.011	0.504 ±0.011
MTASK-M	0.486 ±0.012	0.514 ±0.012

Table 3: Results of human evaluation, showing relative gain of MTASK-S and MTASK-M systems over the LSTM-MMI baseline in pairwise comparison, together with 95% confidence intervals of the means.

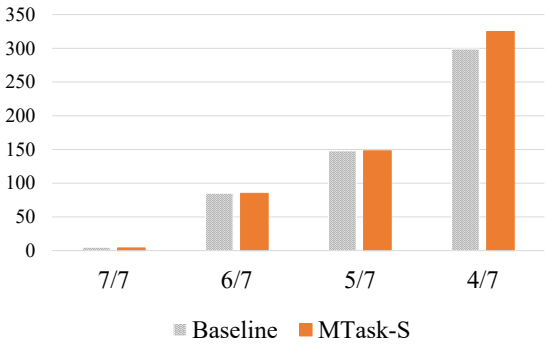

Figure 5: Judge agreement counts for MTASK-S versus Baseline. The difference between the two systems is statistically significant.

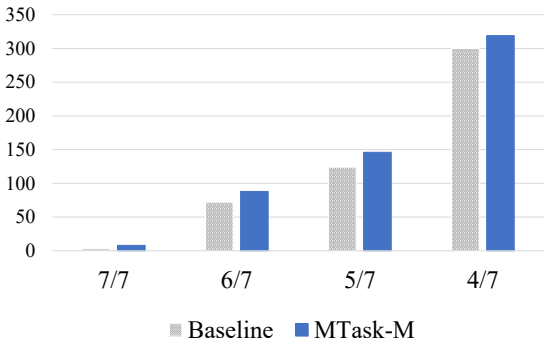

Figure 6: Judge agreement counts for MTASK-M versus Baseline. The difference between the two systems is statistically significant, and the preference for MTASK-M is more evident.

the case where all 7 judges "voted" for the system, bin 6 the case where 6 out of 7 judges "voted" for the system, and so forth.[5] Bins 3 through 0 are not shown since these are a mirror image of bins 7 through 4. It can be seen that judge support for MTASK-M (Figure 6) tends to be stronger than for MTASK-S (Figure 5).

These differences are statistically significant, but they also suggest that this was a challenging task for crowd workers. In many cases, the 5 random examples may not have been sufficed to distinguish individual styles,[6], and even when distinctive, similar outputs from arbitrary inputs may not be undesirable—indeed, different individuals may legitimately respond similarly to the same input, particularly when the input itself is bland or commonplace.

8 Discussion

Fig. 7 presents responses generated by baseline and multi-task (MTASK-M) response generation systems. Both systems are presented with a conversation history of up to two dialog turns (context and input message), and this larger context helps produce responses that are more in line with the conversation flow (Sordoni et al., 2015). The first six response examples are generated for the same underlying speaker (a technical support person, User 1 in Fig. 4). The two last multi-task responses are generated for User 2.

We notice striking differences between the baseline and the multi-task model. The six first responses of **Multitask** in Fig. 7 represent a very consistent register in three different aspects. First, it is relatively clear from these responses that the underlying speaker represented by the model is a tech support employee. Interestingly, this employee appears to give help with fitness-related software, and responds that the customer will be "*able to run a marathon*". On the other hand, the output of the baseline system is relatively bland and deflective. More crucially the baseline does a relatively poor job producing content words that are relevant to the speaker's domain of expertise. Finally, the tone of the baseline system is often incongruous, e.g., when it responds "*I don't know what to do with it*", which is unlikely to be a desirable response to offer a customer. In another case, the baseline responds "*thanks for the info*". While this kind of response is appropriate for many speaker roles, it is less appropriate here, as the support employee is the one to give information and helpful advice.

The figure also illustrates current limitations of our speaker role model. For example, our response in the fourth example shows that such systems can be deflective (e.g., not giving any suggestion in response to "*any idea*"), but at least the system

[5]Partial scores were rounded up. This affects both systems equally.

[6]We limited the number to 5 with the intention of not overwhelming judges with too much information, which may have exacerbated the difficulty.

Context: have you tried uninstalling nike+ connec? how about docking it to a different computer?
Input: Tried both of those techniques. But nothing works. I can try the other computer again.
Baseline: Let me know if there's anything else I can help you with. Thanks for the help.
Multitask: Have you tried using a different browser? Let us know if you have any other questions.

Context: What do you see when you have a failed login attempt? Never ending spinner? Loop you back to login page?
Input: I am getting a loop back to login page, windows.
Baseline: Ah, ok. Thanks for the info. I'll check it out. Thanks for the feedback.
Multitask: I'm sorry to hear that. Have you tried clearing your cache and cookies?

Context: To clarify: your first run after resetting your connection will not post to Facebook. Does this help? Let us know.
Input: That's correct. I reset it twice! It still doesn't work.
Baseline: Let me know if there's anything else I can help you with. Thanks for the update.
Multitask: I'm sorry to hear that. Are you receiving any error messages? Let us know.

Context: To link your sensor to your iPod, from home screen go to settings>Nike+ iPod>sensor>link new. Any luck?
Input: not work.. Any idea?
Baseline: I'm not sure, I'll check it out when i get home. Thanks for the update
Multitask: Let me know if you have any other questions. Let us know if the issue persists.

Context: You're welcome. What generation of the iPod Nano are you using?
Input: 5 years old and bought the kit then but my new Nano touch also not detecting sensor maybe receiver broken
Baseline: Ah , I see. I'll have to check it out. Thanks for the update.
Multitask: Ah, I see. I see. Have you been able to try a different network?

Context: Unfortunately you can't transfer your Nike+ training program to your iPod at this time. Let us know if this helps!
Input: Oh! That's a pity I bet I'd be able to run a couple more if this nice voice, told me to speed up in-between
Baseline: That's what I'm hoping for, but I don't know what to do with it.
Multitask: I'm sure you will be able to run a marathon at least once a week or so.

Context: So when are you going to post more n/d stuff?
Input: Soon I guess I have a chapter written, but it's way too short and not good enough to post just yet.
Baseline: Let me know how you get on with it. I don't know what to write.
Multitask: Ah, I see. Well, I'm sure it'll be interesting to see what happens!

Context: Ah, no. Had kind of a crazy day, and I've been writing a lot tonight. I have a new one-shot almost finished!
Input: Alright. I'm sure you'll read it eventually
Baseline: I don't think I'll be able to sleep tonight, but I'm going to sleep.
Multitask: I hope so! I can't wait to read it, hehe.

Figure 7: Responses produced by our baseline and multi-task neural response generation system (Multi-S), given a conversation history (context and input message.) The first 6 examples and last 2 examples correspond respectively to Users 1 (tech support) and 2.

does respond in a customer-support register. In the fifth example, response of the system seems relatively irrelevant, but this kind of natural language comprehension problem seems almost unavoidable. Semantic congruity aside, the response strikes the right tone—it is pragmatically and socially appropriate, which is the primary purpose of this investigation. The final two examples of Fig. 7 show that the model is also able to learn a voice or register for a completely different kind of character. The underlying person is highly assertive—reflected in their use of exclamation marks—and speaks informally (e.g., "hehe"), in a way the tech support person would typically not.

9 Conclusion

This paper introduces a multi-task learning approach to incorporate speaker role characteristics into conversational models using non-conversational data. Both models presented here are relatively simple to implement, and show significant improvement in perplexity and BLEU score over a baseline system. Overall, MTASK-M is more computationally efficient, and effective in generating speaker-role-specific information, as reflected in human evaluation. Responses generated by these models exhibit a marked ability to capture speaker roles, expressive styles and domain expertise characteristic of the targeted user, without heavy recourse to an individual speaker's conversational data.

Acknowledgements

We thank Marjan Ghazvininejad, John Wieting, Vighnesh Shiv, Mari Ostendorf and Hannaneh Hajishirzi for helpful suggestions and discussions.

References

Rich Caruana. 1998. Multitask learning. In *Learning to learn*, Springer, pages 95–133.

Yun-Nung Chen, Dilek Hakkani-Tür, Gokhan Tur, Jianfeng Gao, and Li Deng. 2016. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding. In *Proc. Interspeech*.

Yun-Nung Chen, Ming Sun, Alexander I Rudnicky, and Anatole Gershman. 2015. Leveraging behavioral patterns of mobile applications for personalized spoken language understanding. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, pages 83–86.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *NIPS*.

Renato De Mori, Frédéric Bechet, Dilek Hakkani-Tur, Michael McTear, Giuseppe Riccardi, and Gokhan Tur. 2008. Spoken language understanding. *IEEE Signal Processing Magazine* 25(3).

Michel Galley, Chris Brockett, Alessandro Sordoni, Yangfeng Ji, Michael Auli, Chris Quirk, Margaret Mitchell, Jianfeng Gao, and Bill Dolan. 2015. deltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets. In *Proc. ACL-IJCNLP*.

Yvette Graham, Timothy Baldwin, and Nitika Mathur. 2015. Accurate evaluation of segment-level machine translation metrics. In *Proc. NAACL-HLT*.

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

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. ICLR*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proc. NAACL-HLT*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proc. ACL*.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proc. EMNLP*.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proc. HLT-NAACL*.

Yi Luan, Yangfeng Ji, Hannaneh Hajishirzi, and Boyang Li. 2016a. Multiplicative representations for unsupervised semantic role induction. In *Proc. ACL*.

Yi Luan, Yangfeng Ji, and Mari Ostendorf. 2016b. LSTM based conversation models. *arXiv preprint arXiv:1603.09457* .

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. *arXiv preprint arXiv:1708.06075* .

Yi Luan, Shinji Watanabe, and Bret Harsham. 2015. Efficient learning for spoken language understanding tasks with word embedding based pre-training. In *Proc. Interspeech*.

Yi Luan, Richard Wright, Mari Ostendorf, and Gina-Anne Levow. 2014. Relating automatic vowel space estimates to talker intelligibility. In *Proc. Interspeech*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *ICLR*.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2017. Coherent dialogue with attention-based language models. In *Proc. AAAI*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*.

Mark Przybocki, Kay Peterson, Sébastien Bronsart, and Gregory Sanders. 2009. The NIST 2008 metrics for machine translation challenge—overview, methodology, metrics, and results. *Machine Translation* 23(2):71–103.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proc. EMNLP*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proc. ACL*.

Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. 2017. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *Proc. EMNLP*.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Meg Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proc. NAACL-HLT*.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *Proc. ICML Deep Learning Workshop*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proc. EMNLP*.