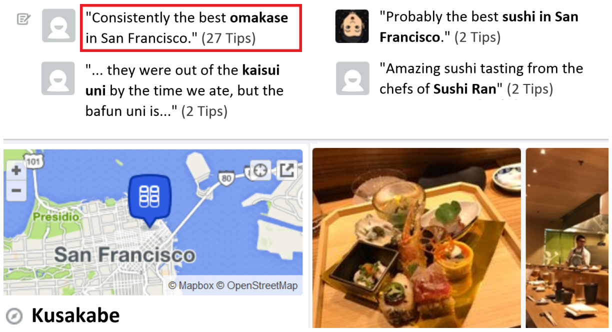
A Knowledge-Grounded Neural Conversation Model[*]

Marjan Ghazvininejad[1†] **Chris Brockett**[2] **Ming-Wei Chang**[2‡]
Bill Dolan[2] **Jianfeng Gao**[2] **Wen-tau Yih**[2§] **Michel Galley**[2]
[1]Information Sciences Institute, USC
[2]Microsoft Research
ghazvini@fb.com, mgalley@microsoft.com

Abstract

Neural network models are capable of generating extremely natural sounding conversational interactions. However, these models have been mostly applied to casual scenarios (e.g., as "chatbots") and have yet to demonstrate they can serve in more useful conversational applications. This paper presents a novel, *fully data-driven*, and knowledge-grounded neural conversation model aimed at producing more contentful responses. We generalize the widely-used Sequence-to-Sequence (SEQ2SEQ) approach by conditioning responses on both conversation history and external "facts", allowing the model to be versatile and applicable in an open-domain setting. Our approach yields significant improvements over a competitive SEQ2SEQ baseline. Human judges found that our outputs are significantly more informative.



"Consistently the best **omakase** in San Francisco." (27 Tips)

"Probably the best **sushi in San Francisco.**" (2 Tips)

"... they were out of the **kaisui uni** by the time we ate, but the bafun uni is..." (2 Tips)

"Amazing sushi tasting from the chefs of **Sushi Ran**" (2 Tips)

User input: *Going to Kusakabe tonight.*
Neural model: Have a great time!
Human: You'll love it! Try omasake, the best in town.

Figure 1: Responses of fully data-driven conversation models are often appropriate, but generally lack content characteristic of human responses.

Introduction

Recent work has shown that conversational chatbot models can be trained in an end-to-end and completely data-driven fashion, without hand-coding (**?**; **?**; **?**; **?**; **?**, *inter alia*). However, fully data-driven systems still lack grounding in the real world and do not have access to external knowledge (textual or structured), which makes it challenging for such systems to respond substantively. Fig. 1 illustrates the difficulty: while an ideal response would directly reflect on the entities mentioned in the query (user input), neural models produce responses that, while conversationally appropriate, seldom include factual content. This contrasts with traditional dialog systems, which can readily inject entities and facts into responses, but often at the cost of significant hand-coding. Slot-filler dialog systems are hard put to come up with a natural sounding utterance like the second response in Fig. 1 in a manner that is generalizable and scalable.

The goal of this work is to benefit from the versatility and scalability of fully data-driven models, while simultaneously seeking to produce models that are usefully grounded in external knowledge, permitting them to be deployed in, for example, recommendation systems (e.g., for restaurants), and to adapt quickly and easily to new domains. The objective here is not task completion as in traditional dialog systems, but the ability to engage a user in a relevant and informative conversation. The tie to external data is critical, as the requisite knowledge is often not stored in conversational corpora. Much of this information is not found in structured databases either, but is textual, and can be mined from online resources such as Wikipedia, book reviews on Goodreads, and restaurant reviews on Foursquare.

This paper presents a novel, *fully data-driven*, knowledge-grounded neural conversation model aimed at producing contentful responses. Our framework generalizes the Sequence-to-Sequence (SEQ2SEQ) approach (**?**; **?**) of previous neural conversation models, as it naturally combines conversational and non-conversational data via techniques such as multi-task learning (**?**; **?**). The key idea is that we can condition responses not only based on conversation history (**?**), but also on external "facts" that are relevant to the current context (for example, Foursquare entries as in Fig. 1). Our approach only requires a way to infuse external information based on conversation context (e.g., via simple entity name matching), which makes it highly versatile and applicable in an open-domain setting. Using this framework, we

[*] This work was conducted at Microsoft Research.
[†] Now at Facebook AI Research.
[‡] Now at Google AI Language.
[§] Now at Allen Institute for Artificial Intelligence.

A: **Looking forward to trying** *@pizzalibretto* **tonight! my expectations are high**.
B: **Get** *the rocco salad*. **Can you eat** *calamari*?

A: **Anyone in** *Chi* **have a** *dentist office* **they recommend? I'm never going back to** *[...]* **and would love a reco!**
B: **Really looved** *Ora* **in** *Wicker Park*.

A: **I'm at** *California Academy of Sciences*
B: **Make sure you catch** *the show* **at** *the Planetarium*. **Tickets are usually limited.**

A: **I'm at** *New Wave Cafe*.
B: **Try to get to** *Dmitri's* **for dinner**. **Their** *pan fried scallops* **and** *shrimp scampi* **are to die for.**

A: **I just bought:** *[...] 4.3-inch portable GPS navigator* **for my wife, shh, don't tell her**.
B: **I heard this brand** *loses battery power*.

Figure 2: Social media datasets include many contentful and useful exchanges, e.g., here recommendation dialog excerpts extracted from real tweets. While previous models (e.g., SEQ2SEQ) succeed in learning the **backbone of conversations**, they have difficulty modeling and producing *contentful words* such as named entities, which are sparsely represented in conversation data. To help solve this issue, we rely on non-conversational texts, which represent such entities much more exhaustively.

have trained systems at a large scale using 23M general-domain conversations from Twitter and 1.1M Foursquare tips, showing significant improvements in terms of informativeness (human evaluation) over a competitive large-scale SEQ2SEQ model baseline. To the best of our knowledge, this is the first large-scale, fully data-driven neural conversation model that effectively exploits external knowledge.

Related Work

The present work situates itself within the data-driven paradigm of conversation generation, in which statistical and neural machine translation models are derived from conversational data (**?**; **?**; **?**; **?**; **?**; **?**). The introduction of contextual models by (**?**) was an important advance within this framework, and we extend their basic approach by injecting side information from textual data. Introduction of side information has been shown to be beneficial to machine translation (**?**), as has also the incorporation of images into multi-modal translation (**?**; **?**). Similarly, (**?**) employ a knowledge graph to embed side information into dialog systems. Multi-task learning can be helpful in tasks ranging from query classification to machine translation (**?**; **?**; **?**; **?**). We adopt this approach in order to implicitly encode relevant external knowledge from textual data.

This work should be seen as distinct from more goal-directed neural dialog modeling in which question-answer slots are explicitly learned from small amounts of crowd-sourced data, customer support logs, or user data (**?**; **?**; **?**; **?**). In many respects, that paradigm can be characterized as the neural extension of conventional dialog models with or without statistical modeling, e.g., (**?**; **?**; **?**; **?**). Our purpose is to explore the space of less clearly goal-directed, but nonetheless informative (i.e., informational) conversation that does not demand explicit slot-filling.

Also relevant is (**?**), who employ memory networks to handle restaurant reservations, using a small number of keywords to handle entity types in a structured knowledge base. Similarly (**?**) use memory networks to manage dialog state.

These works utilize datasets that are relatively small, and unlikely to scale, whereas we leverage free-form text to draw on datasets that are several orders of magnitude larger, allowing us to cover a greater diversity of domains and forms and thereby learn a more robust conversational backbone.

Grounded Response Generation

A primary challenge in building fully data-driven conversation models is that most of the world's knowledge is not represented in any existing conversational datasets. While these datasets (**?**) have grown dramatically in size thanks in particular to social media (**?**), such datasets are still very far from containing discussions of every entry in Wikipedia, Foursquare, Goodreads, or IMDB. This problem considerably limits the appeal of existing data-driven conversation models, as they are bound to respond evasively or deflectively as in Fig. 1, especially with regard to those entities that are poorly represented in the conversational training data. On the other hand, even where conversational data representing most entities of interest may exist, we would still face challenges as such huge dataset would be difficult to apply in model training, and many conversational patterns exhibited in the data (e.g., for similar entities) would be redundant.

Our approach aims to avoid redundancy and attempts to better generalize from existing conversational data, as illustrated in Fig. 2. While the conversations in the figure are about specific venues, products, and services, conversational patterns are general and equally applicable to other entities. The learned conversational behaviors could be used to, e.g., recommend other products and services. A traditional dialog system would use predefined slots to fill conversational backbone (bold text) with content; here, we present a more robust and scalable approach.

In order to infuse the response with factual information relevant to the conversational context, we propose the knowledge-grounded model architecture depicted in Fig. 3. First, we have available a large collection of world facts,[1] which is a large collection of raw text entries (e.g.,

[1] For presentation purposes, we refer to these items as "facts", but a "fact" here is simply any snippet of authored text, which may

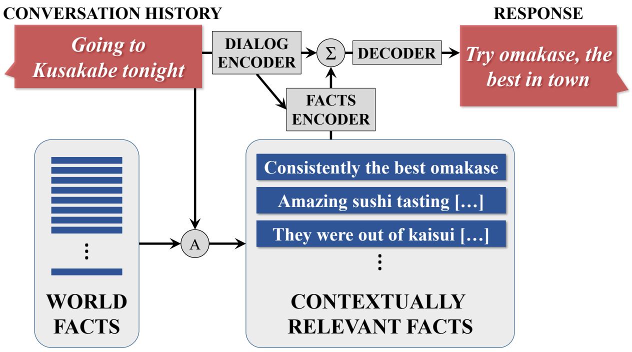

CONVERSATION HISTORY
Going to Kusakabe tonight
DIALOG ENCODER
Σ
DECODER
RESPONSE
Try omakase, the best in town
FACTS ENCODER
A
WORLD FACTS
Consistently the best omakase
Amazing sushi tasting […]
They were out of kaisui […]
CONTEXTUALLY RELEVANT FACTS

Figure 3: Knowledge-grounded model architecture.

Foursquare, Wikipedia, or Amazon reviews) indexed by named entities as keys. Then, given a conversational history or source sequence S, we identify the "focus" in S, which is the text span (one or more entities) based on which we form a query to link to the facts. This focus can either be identified using keyword matching (e.g., a venue, city, or product name), or detected using more advanced methods such as entity linking or named entity recognition. The query is then used to retrieve all contextually relevant facts: $F = \{f_1, ..., f_k\}$.[2] Finally, both conversation history and relevant facts are fed into a neural architecture that features distinct encoders for conversation history and facts. We will detail this architecture in the subsections below.

This knowledge-grounded approach is more general than SEQ2SEQ response generation, as it avoids the need to learn the same conversational pattern for each distinct entity that we care about. In fact, even if a given entity (e.g., @pizzalibretto in Fig. 2) is not part of our conversational training data and is therefore out-of-vocabulary, our approach is still able to rely on retrieved facts to generate an appropriate response. This also implies that we can enrich our system with new facts without the need to retrain the full system.

We train our system using multi-task learning (**?**; **?**) as a way of combining conversational data that is naturally associated with external data (e.g., discussions about restaurants and other businesses as in Fig. 2), and less informal exchanges (e.g., a response to *hi, how are you*). More specifically, our multi-task setup contains two types of tasks:

(1) one purely conversational, where we expose the model without fact encoder to (S, R) training examples, S representing the conversation history and R the response;

(2) the other task exposes the full model with $(\{f_1, \ldots, f_k, S\}, R)$ training examples.

This decoupling of the two training conditions offers several advantages, including: First, it allows us to pre-train the conversation-only dataset separately, and start multi-task training (warm start) with a dialog encoder and decoder that already learned the backbone of conversations. Second, it gives us the flexibility to expose different kinds of conversational data in the two tasks. Finally, one interesting option is to replace the response in task (2) with one of the facts ($R = f_i$), which makes task (2) similar to an autoencoder and helps produce responses that are even more contentful.

Dialog Encoder and Decoder

The dialog encoder and response decoder form together a sequence-to-sequence (SEQ2SEQ model (**?**; **?**), which has been successfully used in building end-to-end conversational systems (**?**; **?**; **?**). Both encoder and decoder are recurrent neural network (RNN) models: an RNN that encodes a variable-length input string into a fixed-length vector representation and an RNN that decodes the vector representation into a variable-length output string. This part of our model is almost identical to prior conversational SEQ2SEQ models, except that we use gated recurrent units (GRU) (**?**) instead of LSTM (**?**) cells. Encoders and decoders in the present model do not share weights or word embeddings.

Facts Encoder

The Facts Encoder of Fig. 3 is similar to the Memory Network model first proposed by (**?**; **?**). It uses an associative memory for modeling the facts relevant to a particular problem—in our case, an entity mentioned in a conversation–then retrieves and weights these facts based on the user input and conversation history to generate an answer. Memory network models have been successfully used in Question Answering to make inferences based on the facts saved in the memory (**?**).

In our adaptation of memory networks, we use an RNN encoder to turn the input sequence (conversation history) into a vector, instead of a bag of words representation as used in the original memory network models. This enables us to better exploit interlexical dependencies between different parts of the input, and makes this memory network model (facts encoder) more directly comparable to a SEQ2SEQ model.

More formally, we are given an input sentence $S = \{s_1, s_2, ..., s_n\}$, and a fact set $F = \{f_1, f_2, ..., f_k\}$ that are relevant to the conversation history. The RNN encoder reads the input string word by word and updates its hidden state. After reading the whole input sentence the hidden state of the RNN encoder, u is the summary of the input sentence. By using an RNN encoder, we have a rich representation for a source sentence.

Let us assume u is a d dimensional vector and r_i is the bag of words representation of f_i with dimension v. Based on (**?**) we have:

$$m_i = Ar_i \tag{1}$$

$$c_i = Cr_i \tag{2}$$

$$p_i = \text{softmax}(u^T m_i) \tag{3}$$

$$o = \sum_{i=1}^{k} p_i c_i \tag{4}$$

$$\hat{u} = o + u \tag{5}$$

contain subjective or inaccurate information.

[2] In our work, we use a simple keyword-based IR engine to retrieve relevant facts from the full collection (see Datasets section).

Where $A, C \in \mathbb{R}^{d \times v}$ are the parameters of the memory network. Then, unlike the original version of the memory network, we use an RNN decoder that is good for generating the response. The hidden state of the RNN is initialized with $\hat{u}$ which is a symmetrization of input sentence and the external facts, to predict the response sentence R word by word.

As alternatives to summing up facts and dialog encodings in equation 5, we also experimented with other operations such as concatenation, but summation seemed to yield the best results. The memory network model of (**?**) can be defined as a multi-layer structure. In this task, however, 1-layer memory network was used, since multi-hop induction was not needed.

Datasets

The approach we describe above is quite general, and is applicable to any dataset that allows us to map named entities to free-form text (e.g., Wikipedia, IMDB, TripAdvisor, etc.). For experimental purposes, we utilize datasets derived from two popular social media services: Twitter (conversational data) and Foursquare (non-conversational data).

Foursquare: Foursquare tips are comments left by customers about restaurants and other, usually commercial, establishments. A large proportion of these describe aspects of the establishment, and provide recommendations about what the customer enjoyed (or otherwise) We extracted from the web 1.1M tips relating to establishments in North America. This was achieved by identifying a set of 11 likely "foodie" cities and then collecting tip data associated with zipcodes near the city centers. While we targeted foodie cities, the dataset is very general and contains tips applicable to many types of local businesses (restaurants, theaters, museums, stores, etc.) In the interests of manageability for experimental purposes, we ignored establishments associated with fewer than 10 tips, but other experiments with up to 50 tips per venue yield comparable results. We limited the tips to those for which Twitter handles were found in the Twitter conversation data.

Twitter: We collected a **23M general dataset** of 3-turn conversations. This serves as a background dataset not associated with facts, and its massive size is key to learning the conversational structure or backbone.

Separately, on the basis of Twitter handles found in the Foursquare tip data, we collected approximately 1 million two-turn conversations that contain entities that tie to Foursquare. We refer to this as the **1M grounded dataset**. Specifically, we identify conversation pairs in which the first turn contained either a handle of the business name (preceded by the "@" symbol) or a hashtag that matched a handle.[3] Because we are interested in conversations among real users (as opposed to customer service agents), we removed conversations where the response was generated by a user with a handle found in the Foursquare data.

Grounded Conversation Datasets

We augment the 1M grounded dataset with facts (here Foursquare tips) relevant to each conversation history. The number of contextually relevant tips for some handles can sometimes be enormous, up to 10k. To filter them for relevance to the input, the system vectorizes the input (as tf-idf weighted word counts) and each of the retrieved facts, and calculates cosine similarity between the input sentence and each of the tips and retains 10 tips with the highest score.

Furthermore, for a significant portion of the 1M Twitter conversations collected using handles found on Foursquare, the last turn was not particularly informative, e.g., when it provides a purely socializing response (e.g., *have fun there*). As one of our goals is to evaluate conversational systems on their ability to produce *contentful* responses, we select a dev and test set (4k conversations in total) designed to contain responses that are informative and useful.

For each handle, we created two scoring functions:

- Perplexity according to a 1-gram LM trained on all the tips containing that handle.
- χ-square score, which measures how much content each token bears in relation to the handle. Each tweet is then scored according to the average content score of its terms.

In this manner, we selected 15k top-ranked conversations using the LM score and 15k using the chi-square score. A further 15k conversations were randomly sampled. We then randomly sampled 10k conversations from these 45K conversations. Crowdsourced human judges were then presented with these 10K sampled conversations and asked to determine whether the response contained actionable information, i.e., did they contain information that would permit the respondents to decide, e.g., whether or not they should patronize an establishment. From this, we selected the top-ranked 4k conversations to be held out as validation set and test set; these were removed from our training data.

Experimental Setup

Multi-Task Learning

We use multi-task learning with these tasks:

- FACTS task: We expose the full model to $(\{f_1, ..., f_n, S\}, R)$ training examples.
- NOFACTS task: We expose the model without fact encoder to (S, R) examples.
- AUTOENCODER task: This is similar to the FACTS task, except that we replace the response with each of the facts, i.e., this model is trained on $(\{f_1, ..., f_n, S\}, f_i)$ examples. There are n times many samples for this task than for the FACTS task.[4]

[3] This mechanism of linking conversations to facts using exact match on the handle is high precision but low recall, but low recall seems reasonable as we are far from exhausting all available Twitter and Foursquare data.

[4] This is akin to an autoencoder as the fact f_i is represented both in the input and output, but is of course not strictly an autoencoder.

The tasks Facts and NoFacts are representative of how our model is intended to work, but we found that the Autoencoder tasks helps inject more factual content into the response. The different variants of our multi-task learned system exploit these tasks as follows:

- Seq2Seq: Trained on task NoFacts with the 23M general conversation dataset. Since there is only one task, it is not *per se* a multi-task setting.
- MTask: Trained on two instances of the NoFacts task, respectively with the 23M general dataset and 1M grounded dataset (but without the facts). While not an interesting system in itself, we include it to assess the effect of multi-task learning separately from facts.
- MTask-R: Trained on the NoFacts task with the 23M dataset, and the Facts task with the 1M grounded dataset.
- MTask-F: Trained on the NoFacts task with the 23M dataset, and the Autoencoder task with the 1M dataset.
- MTask-RF: Blends MTask-F and MTask-R, as it incorporates 3 tasks: NoFacts with the 23M general dataset, Facts with the 1M grounded dataset, and Autoencoder again with the 1M dataset.

We trained a one-layer memory network structure with two-layer Seq2Seq models. More specifically, we used 2-layer GRU models with 512 hidden cells for each layer for encoder and decoder, the dimensionality of word embeddings is set to 512, and the size of input/output memory representation is 1024. We used the Adam optimizer with a fixed learning rate of 0.1. Batch size is set to 128. All parameters are initialized from a uniform distribution in $[-\sqrt{3/d}, \sqrt{3/d}]$, where d is the dimension of the parameter. Gradients are clipped at 5 to avoid gradient explosion.

Encoder and decoder use different sets of parameters. The top 50k frequent types from conversation data is used as vocabulary which is shared between both conversation and non-conversation data. We use the same learning technique as (?) for multi-task learning. In each batch, all training data is sampled from one task only. For task i we define its mixing ratio value of α_i, and for each batch we select randomly a new task i with probability of $\alpha_i / \sum_j \alpha_j$ and train the system by its training data.

Decoding and Reranking

We use a beam-search decoder similar to (?) with beam size of 200, and maximum response length of 30. Following (?), we generate N-best lists containing three features: (1) the log-likelihood $\log P(R|S,F)$ according to the decoder; (2) word count; (3) the log-likelihood $\log P(S|R)$ of the source given the response. The third feature is added to deal with the issue of generating commonplace and generic responses such as *I don't know*, which is discussed in detail in (?). Our models often do not need the third feature to be effective, but—since our baseline needs it to avoid commonplace responses—we include this feature in all systems. This yields the following reranking score:

$$\log P(R|S,F) + \lambda \log P(S|R) + \gamma |R|$$

λ and γ are free parameters, which we tune on our development N-best lists using MERT (?) by optimizing BLEU. To estimate $P(S|R)$ we train a Sequence-to-sequence model by swapping messages and responses. In this model we do not use any facts.

Model	Perplexity	
	General Data	Grounded Data
Seq2Seq	**55.0**	214.4
Seq2Seq-S	125.7	**82.6**
MTask	57.2	82.5
MTask-R	**55.1**	**77.6**
MTask-F	77.3	448.8
MTask-RF	67.2	97.7

Table 1: Perplexity of different models. Seq2Seq-S is a Seq2Seq model that is trained on the NoFacts task with 1M grounded dataset (without the facts).

Model	BLEU	Diversity	
		1-gram	2-gram
Seq2Seq	0.55	4.14%	14.4%
MTask	0.80	2.35%	5.9%
MTask-F	0.48	9.23%	26.6%
MTask-R	**1.08**	7.08%	21.9%
MTask-RF	0.58	**8.71%**	**26.0%**

Table 2: BLEU-4 and lexical diversity.

Evaluation Metrics

Following (?; ?; ?), we use BLEU automatic evaluation. While (?) suggest that BLEU correlates poorly with human judgment at the sentence-level,[5] we use instead corpus-level BLEU, which is known to better correlate with human judgments (?), including for response generation (?). We also report perplexity and lexical diversity, the latter as a raw yet automatic measure of informativeness and diversity. Automatic evaluation is augmented with human judgments of appropriateness and informativeness.

Results

Automatic Evaluation: We computed perplexity and BLEU (?) for each system. These are shown in Tables 1 and 2 respectively. We notice that the Seq2Seq model specifically trained on general data has high perplexity on grounded data.[6] We observe that the perplexity of MTask and MTask-R models on both general and grounded data is as low as the Seq2Seq models that are trained specifically

[5] This corroborates earlier findings that accurate sentence-level automatic evaluation is indeed difficult, even for Machine Translation (?), as BLEU and related metrics were originally designed as corpus-level metrics.

[6] Training the system on just 1M grounded data with Facts doesn't solve this problem, as its perplexity on general data is also high (not in table).

on general and grounded data respectively. As expected, injecting more factual content into the response in MTask-F and MTask-RF increased the perplexity especially on grounded data.

BLEU scores are low, but this is not untypical of conversational systems (e.g., (**?**; **?**)). Table 2 shows that the MTask-R model yields a significant performance boost, with a BLEU score increase of 96% and 71% jump in 1-gram diversity compared to the competitive Seq2Seq baseline. In terms of BLEU scores, MTask-RF improvements is not significant, but it generates the highest 1-gram and 2-gram diversity among all models.

Human Evaluation: We crowdsourced human evaluations. We had annotators judge 500 randomly-interleaved paired conversations, asking them which was better on two parameters: appropriateness to the context, and informativeness. The crowd workers were instructed to: *Decide which response is more appropriate, i.e., which is the best conversational fit with what was said. Then decide which of the two is more informative (i.e., knowledgeable, helpful, specific) about the establishment under discussion.* Judges were asked to select among *Clearly#1*, *Maybe #Number 1*, *About the Same*, *Maybe #2*, and *Clearly #2*. These were converted to scores between 1 and 0, and assigned to the pair members depending on the order in which the pair was presented. Seven judges were assigned to each pair.[7]

The results of annotation are shown in Table 3. Our primary system MTask-R, which performed best on BLEU, significantly outperforms the Seq2Seq baseline on *Informativeness* (p = 0.003) and shows a small, but non-statistically-significant gain with respect *Appropriateness*. Other systems are included in the table for completeness. The “vanilla” MTask shows no significant gain in *Informativeness*. MTask-F performed significantly better than baseline (p = 0.005) on *Informativeness*, but was significantly worse on *Appropriateness*. MTask-RF came in slightly better than baseline on *Informativeness* but worse on *Appropriateness*, though in neither case is the difference statistically significant by the conventional standard of $\alpha = 0.05$. In sum, our best performing MTask-R system appears to have successfully balanced the needs of informativeness and maintaining contextual appropriateness.

The narrow differences in averages in Table 3 tend to obfuscate the judges' voting trends. To clarify the picture, we translated the scores into the ratio of judges who preferred that system and binned the counts. Figs. 4 and 5 compare MTask-R with the Seq2Seq baseline. Bin 7 on the left corresponds to the case where all 7 judges “voted” for the system, bin 6 to that where 6 out of 7 judges “voted” for the system, and so on.[8] Other bins are not shown since these are a mirror image of bins 7 through 4. The distributions in Fig. 5 are sharper and more distinctive than in Fig. 4. indicating that judge preference for the MTask-R model is relatively stronger when it comes to informativeness.

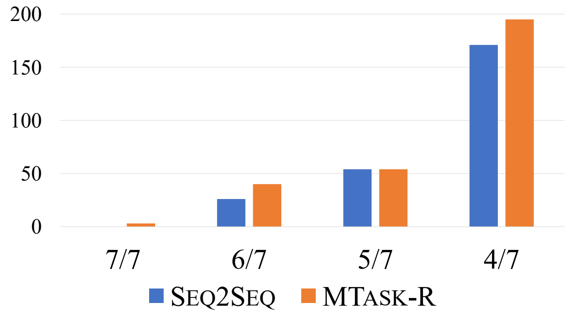


Figure 4: Judge preference counts (appropriateness) for MTask-R versus Seq2Seq.

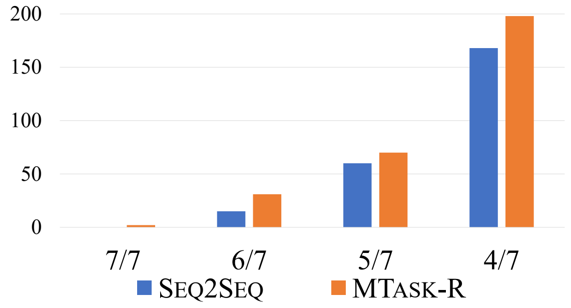


Figure 5: Judge preference counts (informativeness) for MTask-R versus Seq2Seq.

Discussion

Figure 6 presents examples from the MTask-RF model, and illustrates that our responses are generally both appropriate and informative. First, we note that our models preserve the ability of earlier work (**?**; **?**) to respond contextually and appropriately on a variety of topics, with responses such as *me too* (1) and *have a safe flight* (2). Second, our grounded models often incorporate information emerging from “facts”, while usually keeping the responses contextually appropriate. For example in (3), those facts revolve mostly around jewelry such as bracelets, which leads the system to respond *obsessed with their bracelets and the meaning behind them*, while *meaning behind them* does not belong to any of the facts and is instead “inferred” by the model (which associates jewelry with sentimental value). Responses influenced mostly by facts may occasionally contain a single unedited fact (4-5), but otherwise generally combine text from different facts (6-8).[9] Examples 9 and 10 are negative examples that illustrate the two main causes of system errors: the presence of an irrelevant fact (e.g., *j.crew* in example 9), and the system combining self-contradictory facts (10). Despite such errors, judges found that our best grounded system is generally on par with the Seq2Seq sys-

[7] Annotators whose variance fell greater than two standard deviations from the mean variance were dropped.

[8] Partial scores were rounded up, affecting both systems equally.

[9] Facts: *grab a cup of coffee and get productive* and *try the cookies in the vending machine of local food* (6); *sit with and take a picture of the Simpsons on the 3rd floor* and *Check out the video feed on 6 and Simpsons/billiards on 3!* (8).

Comparison	Informativeness				Appropriateness			
Seq2Seq vs MTask	0.501	±0.016	0.499	±0.015	**0.530**	**±0.017**	0.470	±0.017
Seq2Seq vs MTask-F	0.478	±0.015	**0.522**	**±0.015**	**0.537**	**±0.016**	0.463	±0.017
Seq2Seq vs MTask-RF	0.492	±0.013	0.508	±0.013	0.502	±0.015	0.498	±0.014
Seq2Seq vs MTask-R (*)	0.479	±0.017	**0.521**	**±0.013**	0.495	±0.015	0.505	±0.015

Table 3: Mean differences in judgments in human evaluation, together with 95% confidence intervals. Differences sum to 1.0. Boldface items are significantly better (p <0.05) than their comparator. (*): Main system, pre-selected on the basis of BLEU.

1	A: I would give ALMOST anything for some [*Mexican restaurant*] right now. B: Me too. Creamy jalapeno sauce is really good.
2	A: Visiting the celebs at Los Angeles International Airport (LAX) - [...] w/ 70 others B: Nice airport terminal. Have a safe flight.
3	A: Obsessed with [*jewelry company*] :-* B: oh my gosh obsessed with their bracelets and the meaning behind them !
4	A: Live right now on [*FM radio*] Tune in!!!!! B: Listen to Lisa Paige
5	A: Is that [...] in your photos? It's on my list of places to visit in NYC. B: Don't forget to check out the 5th floor, while you are here, it's a great view.
6	A: Are you [*coworking community*] today or tomorrow? B: I think so. Try the cookies and coffee.
7	A: Cant wait for my [*pizzeria*] tonight B: Order bacon n pepperoni thin crust pizza.
8	A: So [...] is down to one copy of Pound Foolish. I'm curious to see if they are re-ordering it. B: Check out the video feed on 6 and take a picture of the Simpsons on the 3rd floor.
9	A: I wish [...] would introduce another vegetarian option besides the shroomburger. It's delicious but kind of ridiculous. B: This is the best j.crew in the world. Try the lemonade!
10	A: Just had an awesome dinner at [*restaurant*] Great recommendation [*person*] B: One of my favorite places I've ever been to in NYC. The food is great and the service is lackluster.

Figure 6: Responses produced by our neural response generation system MTask-RF. Entities marked [...] are anonymized.

tem in terms of appropriateness, while significantly improving informativeness (Table 3).

Conclusions

We have presented a novel knowledge-grounded conversation engine that could serve as the core component of a multi-turn recommendation or conversational QA system. The model is a large-scale, scalable, fully data-driven neural conversation model that effectively exploits external knowledge, and does so without explicit slot filling. It generalizes the Seq2Seq approach to neural conversation models by naturally combining conversational and non-conversational data through multi-task learning. Our simple entity matching approach to grounding external information based on conversation context makes for a model that is informative, versatile and applicable in open-domain systems.

Acknowledgements

We thank Xuetao Yin and Leon Xu for helping us obtain Foursquare data, and Kevin Knight, Chris Quirk, Nebojsa Jojic, Lucy Vanderwende, Vighnesh Shiv, Yi Luan, John Wieting, Alan Ritter, Donald Brinkman, and Puneet Agrawal.

Provident quae amet distinctio quaerat necessitatibus, placeat ducimus dolores neque magni perspiciatis.Sint quae asperiores excepturi odit quidem quo suscipit quod officia numquam aperiam, vero sint officiis autem neque voluptate nobis beatae ducimus inventore velit, illo maxime saepe corporis nulla ad, reprehenderit ducimus id excepturi laudantium eveniet architecto?Iste adipisci officiis optio numquam maiores facere molestias distinctio sed, commodi beatae accusamus sequi nemo sed nihil reprehenderit voluptate consequuntur.In quo odio exercitationem consectetur sit quasi obcaecati at laborum sint, itaque animi tenetur omnis dignissimos perferendis tempora qui dolor totam cupiditate unde, repudiandae in quia ratione saepe?Accusantium fuga inventore accusamus eius qui nostrum reiciendis, pariatur obcaecati dolor laudantium rerum assumenda ut rem vitae temporibus, illo vero ad explicabo nihil laudantium reiciendis animi sunt vitae beatae?Nesciunt veritatis debitis repellendus vel commodi provident, et beatae nesciunt tenetur aliquam fuga aperiam similique, omnis porro odit consectetur quos consequuntur.Recusandae nulla quibusdam cupiditate autem unde temporibus nisi, aspernatur ut iste dolorum corporis, non ea ad dolore velit voluptatibus deserunt neque consequuntur tempore nesciunt.