May I take your order? A Neural Model for Extracting Structured Information from Conversations

Baolin Peng[1], **Michael L. Seltzer**[2],
Yun-Cheng Ju[2], **Geoffrey Zweig**[2] **and Kam-Fai Wong**[1]

[1]The Chinese University of Hong Kong, Shatin, N.T. Hong Kong
[2]Microsoft Research, One Microsoft Way, Redmond, WA, USA
{blpeng, kfwong}@se.cuhk.edu.hk
{mseltzer, yuncj, gzweig}@microsoft.com

Abstract

In this paper we tackle a unique and important problem of extracting a structured order from the conversation a customer has with an order taker at a restaurant. This is motivated by an actual system under development to assist in the order taking process. We develop a sequence-to-sequence model that is able to map from unstructured conversational input to the structured form that is conveyed to the kitchen and appears on the customer receipt. This problem is critically different from other tasks like machine translation where sequence-to-sequence models have been used: the input includes two sides of a conversation; the output is highly structured; and logical manipulations must be performed, for example when the customer changes his mind while ordering. We present a novel sequence-to-sequence model that incorporates a special attention-memory gating mechanism and conversational role markers. The proposed model improves performance over both a phrase-based machine translation approach and a standard sequence-to-sequence model.

1 Introduction

Extracting structured information from unstructured text is a critically important problem in natural language processing. In this paper, we attack a deceptively simple form of the problem: understanding what a customer wants when ordering at a restaurant. In this problem, we seek to convert the conversation between the customer and the order taker, i.e. the waiter or waitress, into the structured form that is conveyed to the kitchen to prepare the food, and which appears on the customer receipt.

Waiter:	Hi, how can I help you ?
Customer:	We'd like a large cheese pizza.
Waiter:	Any toppings?
Customer:	Yeah, how about pepperoni and two diet cokes.
Waiter:	What size?
Customer:	Uh, medium and make that three cokes.
Waiter:	Anything else?
Customer:	A small Caesar salad with the dressing on the side
Waiter:	Sure, is that it?
Customer:	Yes, that's all, thanks.

Figure 1: A conversation example of an order-taking interaction at a restaurant.

Item	Size	Qty	Modifiers
Pizza	large	1	add pepperoni
Caesar Salad	small	1	side dressing
Diet Coke	medium	3	

Table 1: An example of the structured data record corresponding to the conversation in Figure 1

We develop this system to analyze real-time interactions with the aim of discovering errors in the order-entry process. Note that the objective is to analyze the interaction and suggest corrections to the human order-taker. Thus, we take both sides of the order-taking interaction as input, and are not attempting to predict the order-taker's side of the conversation.

While we focus on the restaurant domain in this work, this problem is relevant in any scenario in which a conversation results in the creation of structured information. Other examples include a sales interaction which results in a purchase order, a call to a help desk which results in a service record, or a conversation with a travel agent that results in an itinerary.

An example of the problem of interest is shown in Figure 1. The structured data record that corresponds to this conversation is shown in Table 1. There are several things to note about this example:

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 450–459,
Valencia, Spain, April 3-7, 2017.

- The output is a stylized and structured representation of the input
- The items in the structured order may appear in a different sequence than they are mentioned
- Inference occurs across turns, for example that "medium" applies to the coke and not the pizza whose size was earlier specified
- Logical manipulations must be done, for example changing the number of cokes from two to three
- In contrast to machine translation, we do not wish to create a verbatim "translation" of the input, but instead a logical distillation of it

To attack this problem, we implemented two baselines and several sequence-to-sequence models. The first baseline is an information-retrieval approach based on a TF-IDF match (Salton et al., 1975) which finds the most similar conversation in the training data, and returns the associated order. The second uses phrase-based machine translation (Koehn et al., 2003) to "translate" from the conversational input to the tokens in the structured order. We compare these to a sequence-to-sequence (s2s) model with attention (Chan et al., 2016; Bahdanau et al., 2014; Devlin et al., 2015; Yao and Zweig, 2015; Sutskever et al., 2014; Mei et al., 2016), and then extend the s2s model with the addition of a gating mechanism on the attention memory and with an auxiliary input that indicates the conversational role of the speaker (customer or order-taker). We show that it is in fact possible to extract the orders from conversations recorded at a real restaurant [1], and achieve an F measure of over 70 from raw text and 65 from ASR transcriptions.

2 Problem Formulation

The precise problem setting in this paper is as follows. The training data consists of input/output pairs of examples $(X_1, Y_1), \ldots, (X_N, Y_N)$, where X_k is a conversation consisting of several utterances, similar to the example shown in Figure 1, and Y_k is the corresponding structured data record such as the one in Table 1.

[1]The restaurant will remain anonymous for business reasons, and we have changed the names of menu items in our examples accordingly.

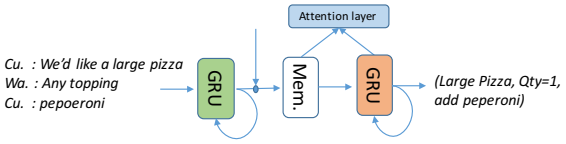


Figure 2: An input unstructured conversation and the corresponding structured record.

Given a conversation X_k, the goal of our model is to extract the structured data record Y_k so that:

$$Y_k = \underset{Y}{\operatorname{argmax}} \log P(Y|X_k) \quad (1)$$

We cast this task as a sequence modeling problem which aims to map the sequence of words in a conversation X_k to the sequence of tokens in the corresponding structured data record Y_k. The input sequence is formed by concatenating the utterances in the conversation, while the output sequence is formed by concatenating the rows in the structured data record. For example, the utterances in the conversation shown in Figure 1 are concatenated to predict the sequence y = *Pizza, size=large, qty=1, modifiers=(add pepperoni) | Diet Coke, size=medium, qty=3 | Caesar Salad, size=small, qty=1, modifiers=(side dressing)* which is derived from Table 1. Under this sequential model, the conditional probability of the structured data record Y given the observed conversation X can be written as

$$P(Y|X,\theta) = \prod_{t=1}^{T} P(y_t|y_{1:t-1}, X, \theta) \quad (2)$$

where $y_{1:t-1}$ denotes the first $t-1$ terms in the structured data record and θ represents the model parameters.

3 Model

The proposed model is based on an encoder-decoder architecture with attention (Bahdanau et al., 2014), as shown in Figure 2. The encoder network reads the input conversation $\mathbf{X}$ one word at a time and updates its hidden state h_t according to current input w_t and previous hidden state h_{t-1},

$$h_t = f_e(w_t, h_{t-1}), t \in \{1, \cdots, M\} \quad (3)$$

where f_e is a nonlinear function which is elaborated in the following section. After reading all the tokens, the encoder network yields a context

vector c as the representation of the entire conversation.

The decoder then processes this representation and generates a hypothesized structured data record Y as an output sequence, word by word given the context vector c and all previous predicted tokens. The conditional probability can be expressed as follows:

$$P(y_t|y_1, \cdots, y_{t-1}, \mathbf{X}) = f_d(y_{t-1}, s_t, c) \quad (4)$$

$$s_t = g(y_{t-1}, s_{t-1}, c) \quad t \in \{1, \cdots, N\} \quad (5)$$

where f_d and g are nonlinear functions and s_t is the hidden state of decoder at time t. Critically, our decoder also utilizes an attention mechanism, which stores the intermediate encoder representations of each input word for use by the decoder.

Two improvements to the conventional encoder-decoder model architecture are proposed in this work. First, we incorporate gates controlled by the encoder into the neural attention memory to adaptively modulate the representations in the memory based on their semantic importance. Second, we propose a way to incorporate conversational role information into the model to reflect the fact that different participants in a multi-party interaction have different roles and the meaning of certain utterances may be dependent on the speaker's role.

A detailed illustration of the proposed model is shown in Figure 3. We elaborate on each component of this model in the following sections.

3.1 Encoder Network

The encoder network is designed to generate a semantically meaningful representation of unstructured conversations. Several neural network architectures have been proposed for this purpose, including CNNs (Kalchbrenner et al., 2014; Hu et al., 2014), RNNs (Sutskever et al., 2014) and LSTMs (Hochreiter and Schmidhuber, 1997). In this work, we use an encoder constructed from a recurrent neural network with gated RNN units (GRU) (Cho et al., 2014). The GRU has been shown to alleviate the gradient vanishing problem of RNNs, enabling the model to learn long term dependencies in the input sequence. GRUs have been shown to perform comparably to LSTMs (Chung et al., 2014).

At time t, the new state of a GRU is computed as follows:

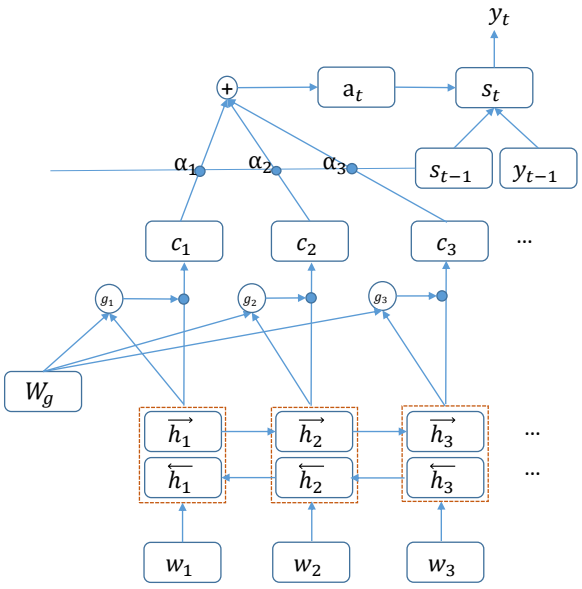


Figure 3: Graphical structure of memory-gated encoder-decoder model with attention mechanism. w_1 represents input; $\overrightarrow{h_1}$ and $\overleftarrow{h_1}$ are the hidden states of forward and backward GRUs, respectively. g_1, α_1 represent the context gates and attention weights, respectively. Small dot node means element-wise product.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (6)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (7)$$

$$\widehat{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1})) \quad (8)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widehat{h}_t \quad (9)$$

where $\odot$ stands for element-wise multiplication. W, U are weight matrixes applied to input and previous hidden state, respectively .h_t is a linear combination of the previous state h_{t-1} and the hypothesis state $\hat{h}_t$. $\hat{h}_t$ is computed with new sequence information. The update gate, z_t, controls to what extent the past information is kept and how much new information is added. The reset gate, r_t, controls to what extent the history state contributes to the hypothesis state. If r_t is zero, then GRU ignores all the history information.

The conversation encoding is obtained by concatenating the GRU hidden state vectors from the forward and backward directions. Thus the encoder operation can be summarized as follows

$$x_t = W_e w_t, t \in [1, T] \quad (10)$$

$$\overrightarrow{h_t} = \overrightarrow{GRU}(x_t), t \in [1, T] \quad (11)$$

$$\overleftarrow{h_t} = \overleftarrow{GRU}(x_t), t \in [T, 1] \quad (12)$$

$$h_t^+ = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \quad (13)$$

where w_t is the one-hot input vector, W_e is the embedding matrix, and x_t is the word embedding for w_t. The functions $\overrightarrow{GRU}(x_t)$ and $\overleftarrow{GRU}(x_t)$ represent the GRU operating in the forward and backward directions, respectively, with processing defined by equations 6–9.

This produces a sequence of context vectors, h_t^+ which are subsequently consumed by an attention mechanism in the decoder. We use the final attention vector h_T^+ to initialize the hidden state of the decoder.

3.2 Memory Gate

In most sequence-to-sequence tasks such as machine translation, every word in the input is important. However, in our scenario, where the input to the system is conversational speech, not all the words in the conversation contribute to the prediction of structured data record. For example, it is reasonable to ignore the chit-chat that is present in many conversations. Further, in other tasks, gating mechanisms have been shown to be useful to dynamically select important information (Yao et al., 2015; Hochreiter and Schmidhuber, 1997; Tu et al., 2016).

In light of this, we propose the use of an additional memory gate to select important information from the memory vector. The memory gate we use consists of a single-layer feed-forward neural network

$$g_t = \sigma(W_g h_t^+ + b_g) \tag{14}$$

where σ is a sigmoid activation function and W_g and b_g are weight matrix and bias, respectively, and h_t^+ is the context vector at time t defined in equation 10. The gate is then applied to the context vector h_t^+ using an element-wise multiplication operation.

$$c_t = g_t \odot h_t^+ \tag{15}$$

After applying memory gate, the gated context vector c_t is then fed into attention memory of the decoder network in place of the original context vector h_t^+. Figure 4 illustrates an example of the gating weights for a sample utterance. The darker colors indicates values close to 1 while the lighter colors indicate values close to 0. As the figure shows, the network learns to suppress semantically unimportant words.

3.2.1 Role Information

In many sequence-to-sequence models, there is no notion of different speakers with different roles. Inspired by the work in dialog generation (Li et al., 2016) and spoken language understanding (Hori et al., 2016), we propose the addition of speaker information into the encoder network to explicitly model the interaction patterns of the customer and order-taker.

Specifically we learn separate word and role embeddings, and concatenate them to form the input. The input to the encoder network becomes:

$$x_t^w = W_e w_t, t \in [1, T] \tag{16}$$
$$x_t^r = W_r r_t, t \in [1, T] \tag{17}$$
$$x_t = x_t^w \oplus x_t^r, t \in [1, T] \tag{18}$$

3.3 Decoder Network

The decoder network is used to predict the next word given all the previously predicted words and the context vectors from the encoder network (Luong et al., 2015; Bahdanau et al., 2014).

We use an RNN with GRU units to predict each word y_t sequentially based on the previously predicted word y_{t-1} and the output of the attention process a_t that computes a weighted combination of the context vectors in memory.

If we define s_t as the hidden layer of the decoder at time t, the decoder's operation can be expressed as

$$s_t = \overrightarrow{GRU}(y_{t-1} \oplus a_t) \tag{19}$$
$$y_t = \text{softmax}(W_o s_t + b_o) \tag{20}$$

where $y_{t-1} \oplus a_t$ is the concatenation of the previously predicted output y_{t-1} and the output of the attention process a_t, and $\overrightarrow{GRU}(\cdot)$ is defined by equations 6–9, as before.

The attention vector a_t is computed as a linear combination of the gated context vectors generated by the encoder network. This can be written as

$$a_t = \sum_{j=1}^{M} \alpha_{ij} c_j \tag{21}$$

where the weights α_{ij} are computed as

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{N} \exp(e_{ik})} \tag{22}$$

A single-layer feed-forward neural network is used to compute e_{ij} as

$$e_{ij} = V_a^T \tanh(W_a s_{t-1} + U_a c_j) \quad (23)$$

where V_a, W_a, and U_a are weight matrices.

3.4 Model Training

The model is trained to maximize the log probability of the structured data records given the corresponding conversation,

$$\sum_{(Y_k, X_k) \in D} \log P(Y_k | X_k) \quad (24)$$

where D is the set containing all the training pairs and $P(Y_k|X_k)$ is computed with equation 2. The standard adadelta algorithm (Zeiler, 2012) is used for parameter updates. Gradients are clipped to 1 to avoid exponentially increasing values (Pascanu et al., 2013).

4 Experiments

In this section, we evaluate our proposed model on two data sets and compare performance with several baseline systems.

4.1 Data sets

We conducted experiments on a corpus of conversations between a customer and an order taker (waiter or waitress) captured in a real restaurant environment. The conversations were manually transcribed by professional annotators. There are 4823 examples in the training set, 543 in the development (dev) set, and 843 in the test set. There are approximately 260 unique items in the record and 150 unique modifiers on these items, but not all modifiers apply to all items. We experimented with two version of the dev and test sets. The first is manually transcribed in the same manner as the training set, while the second is generated by a speech recognition decoder that was trained on the conversations in the training set. We denote the second set as *ASR-dev* and *ASR-test*. Table 2 lists the statistics of the data sets. Note that the audio of a conversation was collected as a single file and then automatically segmented into turns for ASR decoding. This process was not perfect and likely introduced some errors. Thus, the average length and number of turns of differ between the ASR transcriptions and the manual transcriptions.

Data Set	Avg Turns	Avg Length	Avg # Items
Training	8.8	53.5	3.7
Dev	9.7	57.07	3.8
Test	8.7	50.96	3.5
ASR-Dev	8.5	49.8	3.8
ASR-Test	7.8	44.7	3.5

Table 2: Statistics of the experimental corpus. The table denotes the average number of utterances in a conversation, average length of a conversation in words, and average number of items in an order.

4.2 Experimental setup

All words are lower-cased and an unknown word token is used for words which appear less than four times in the training set. The word embedding matrix is initialized by randomly sampling from a normal distribution, and scaled by 0.01. The recurrent connections of the GRU are initialized with orthogonal matrices (Saxe et al., 2013) and biases are initialized to zero. A single layer GRU is used for both the encoder and decoder. The network has 600 hidden units and uses 300-dimensional word embeddings. The dropout rate is set to 0.5. We did not tune hyper-parameters except for the dimension of the role embedding which is selected from $\{3, 5, 10\}$ on the dev set. During inference, we use beam search decoding with a beam of 5 to generate the structured records. In order to decode without a length bias, the log probability of decoded results is normalized by the number of tokens.

4.3 Evaluation

A typical metric to evaluate a generation system is BLEU score (Papineni et al., 2002) which uses ngram overlap to quantify the degree to which a hypothesis matches the reference. However, our scenario is more demanding: order items are either correct or incorrect. Therefore, we adopt precision and recall at the item level as our evaluation metric. Note that an item is defined as a row in the structured data record and typically includes multiple fields. Using Table 1 as an example, there are three items to be scored. Only when the model produces an item that is exactly the same as the reference item do we count it as correct. As an additional measure, we report accuracy of the entire order, in which every item in an order must be correct for the order to be counted as correct.

Model			Dev				Test			
	Gate	Role	Recall	Prec	F1	Accy	Recall	Prec	F1	Accy
IR	-	-	26.5	22.7	24.5	11.4	29.7	25.6	27.5	14.1
PBMT	-	-	64.4	19.3	35.2	29.3	62.6	20.8	36.0	28.4
NAM	-	-	64.9	70.9	67.9	45.7	68.4	71.3	69.8	48.1
NAM	-	✓	65.6	71.6	68.6	45.3	68.8	72.9	70.8	49.1
NAM	✓	-	**67.6**	**72.7**	**70.1**	46.2	68.3	**74.1**	71.1	48.5
NAM	✓	✓	66.9	71.6	69.2	**48.8**	**70.2**	72.3	**71.2**	**51.8**

Table 3: Results of different methods on dev and test set. Human transcriptions are used.

Model			Dev				Test			
	Gate	Role	Recall	Prec	F1	Accy	Recall	Prec	F1	Accy
IR	-	-	21.9	18.9	20.3	6.9	25.7	19.3	23.8	10.2
PBMT	-	-	56.8	20.4	34.1	23.3	56.9	21.5	35.0	24.7
NAM	-	-	56.7	63.5	60.0	36.9	60.3	66.7	63.4	40.6
NAM	-	✓	57.1	64.7	60.8	38.1	62.5	67.4	64.9	42.5
NAM	✓	-	57.0	64.6	60.7	39.2	60.3	68.3	64.2	40.8
NAM	✓	✓	**58.5**	**65.2**	**61.8**	**40.5**	**63.0**	**68.4**	**65.7**	**45.9**

Table 4: Results of different methods on ASR-dev and ASR-test set.

4.4 Baseline systems

We compare the performance of our neural model with baseline models that employ information retrieval (IR) and phrase-based machine translation (PBMT) approaches.

IR: The IR method treated the training set of transcriptions as a collection of documents, each mapped to a corresponding order. The test conversation was used as a query to find the most similar training set conversation. The corresponding order was returned as the estimated order. In our experiment, we use TFIDF to compute the similarity score.

PBMT: The goal of a phrase-based translation model is to map a conversation into its structured record with alignment and language models. In our experiments, we use the Moses decoder, a state-of-the-art phrase-based MT system available for research purposes. We use GIZA++ (Och and Ney, 2003) to learn word alignment and irstlm to learn the language model. The models are trained on the conversation/order pairs in the training set and used to predict the structured data record given a conversation.

4.5 Results

First we discuss the performance of our models on manually transcribed data and then examine the results on ASR recognized data. Table 3 lists the experiment results on manually transcribed dev and test sets. We refer to our model as the neural attention model (NAM). We see that the NAM is superior to both the IR and PBMT methods by a large margin. Both the proposed memory gate and role modifications yield improvements over the basic NAM. When combined, these produce the best performance in terms of accuracy on the dev set, and both F1 and accuracy on the test set. While there are only small differences in the scores among some of the NAM methods, we are unaware of a measure of statistical significance suitable for this task.

Though not reported, we also found that a basic encoder-decoder s2s model without attention performs poorly; it cannot summarize information across multiple turns into a single vector. The attention mechanism, acting on the entire encoding sequence, is critical in our task.

Table 4 shows results on the ASR-dev and ASR-test sets. These data sets are quite noisy since the speech recognizer in this domain has a word error

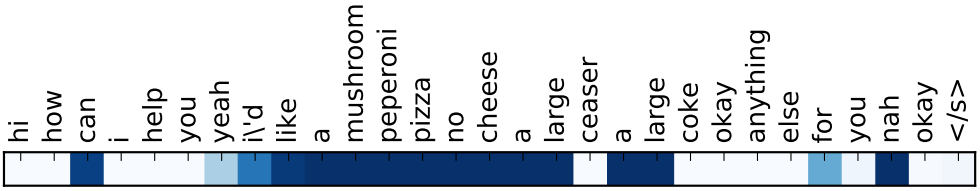
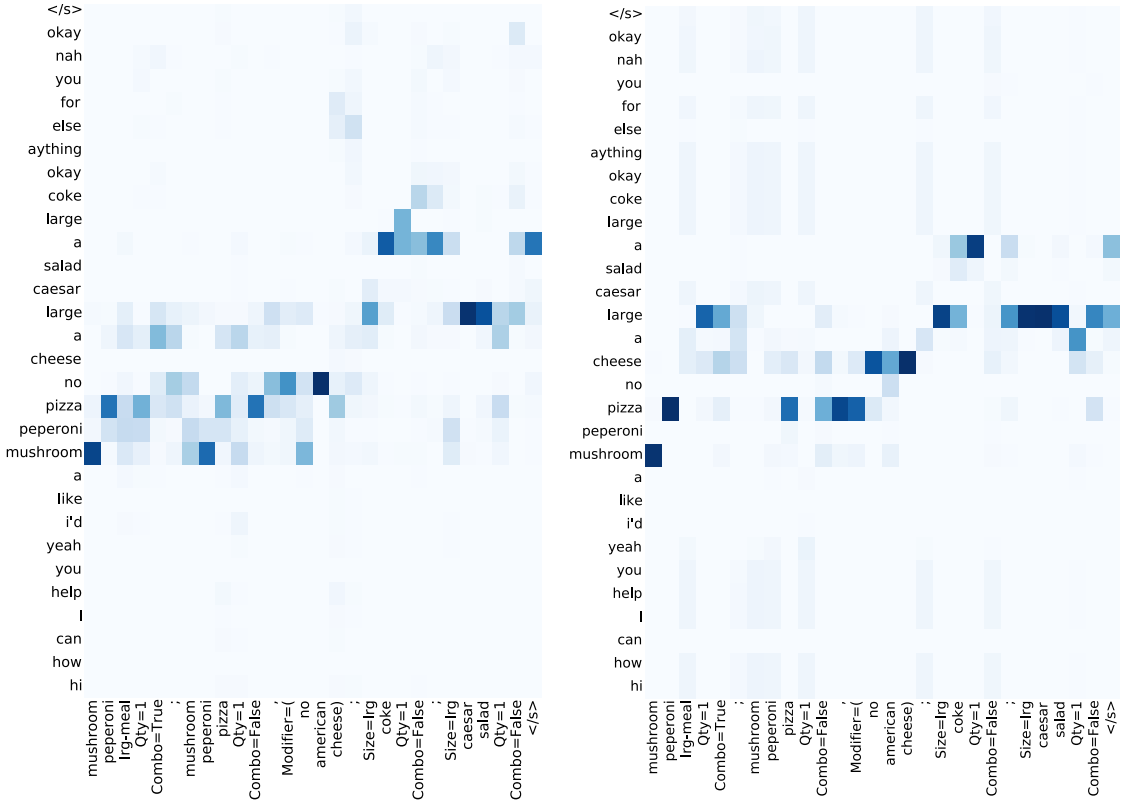


Figure 4: Example of memory gate weights at each time stamp.



(a) Attention weight of NAM

(b) Attention weight of NAM with memory gate

Figure 5: Examples of attention weights of models (a) without memory gate and (b) with memory gate. (b) shows sparse and more focused attention weights. (Better viewed in color.)

rate around 25%. With this noisy data, we find that the memory gate and role additions consistently improve performance. When combined, both F1 and accuracy improved.

4.6 Qualitative analysis

Figure 6 shows a sample input and the output from each model. We see that the NAM augmented with memory gates and role information successfully captures the interaction and generates the correct record.

To better understand the proposed model, we visualize the attention weight at each time step in Figure 5. The figure compares the attention weights produced by a conventional context memory and the proposed gated context memory. We see that both models are able to learn good soft alignment between the input conversation and the output structured data record. However, the attention weights in 5(b), with our proposed gated attention mechanism, are sparser than those in 5(a) and better able to ignore uninformative terms in the input.

5 Related Work

There has been much work on information extraction from single utterances. Kate and Mooney (2006) proposed the use of SVM classifiers based on string kenels to parse natural language to a formal meaning representation. Wong and

Waiter: *My name's Alexis how can I help you.*
Customer: *Yeah can I get a cheese pizza combo?*
Waiter: *Something to drink?*
Customer: *Change that to a mushroom pizza combo with a large sprite no ice.*
Waiter: *Okay.*
Customer: *Thank you.*
Waiter: *Thank you.*

NAM:
Cheese Pizza, Qty=1, Combo=True
Sprite, Size=Lrg, Qty=1, Modifier=(no ice);
Caesar Salad, Size=Med, Qty=1

NAM + Memory Gate:
Cheese Pizza, Qty=1, Combo=True
Sprite, Size=Lrg, Qty=1, Modifier=(no ice);
Caesar Salad, Size=Med, Qty=1
Mushroom Pizza, Qty=1, Combo=True
Sprite, Size=Lrg, Qty=1, Modifier=(no ice);
Caesar Salad, Size=Med, Qty=1

NAM + Memory Gate + Role:
Mushroom Pizza, Qty=1, Combo=True
Sprite, Size=Lrg, Qty=1, Modifier=(no ice);
Caesar Salad, Size=Med, Qty=1

Figure 6: Examples of outputs generated by each model for the conversation in first row.

Mooney (2006) used syntax-based statistical machine translation method to do semantic parsing. Translation of natural language to a formal meaning representation is captured by a synchronous context-free grammar in (Wu, 1997; Chiang et al., 2006). Quirk et al. (2015) created models to map natural language descriptions to executable code using productions from the formal language. Beltagy and Quirk (2016) improved the performance of semantic parsing on If-Then statements by using neural networks to model derivation trees and leveraged several techniques like synthetic training data from paraphrases and grammar combinations to improve generalization and reduce overfitting. In addition, there are some other research works focusing on text generation from structured data records. Angeli et al. (2010) proposed of a domain independent probabilistic approach to performing content selection and surface realization, making text generation as a local decision process. Konstas and Lapata (2013) created a global model to generate text from structured records, which jointly modeled content selection and surface realization with a probabilistic context-free grammar. In contrast, in this paper we focus on generating structured data records from text descriptions.

Using spoken language understanding techniques, (Mesnil et al., 2015) tag each word in a sentence with a predefined slot. A dialog modeling approach (Young et al., 2013) is also relevant to our task. However, this approach requires the definition of semantic slot names and human labeling of dialog acts in each utterance.

There are a number of relevant applications of neural attention models. Nallapati et al. (2016) proposed using sequence to sequence model to summarize source code into natural language; they used a LSTM as encoder and another attentional LSTM and decoder to jointly learn content selection and realization. Dong and Lapata (2016) presented a sequence to sequence model with a tree structure decoder to map natural language to its logical form. The tree structure decoder shows superior performance on data that has nested output structure. It has also been used in other domains including machine translation (Sutskever et al., 2014; Bahdanau et al., 2014), and image caption generation (Fang et al., 2015). From this perspective, the most related work is (Mei et al., 2016) in which they proposed using a sequence-to-sequence model to map navigational instructions in natural language to actions, which is conceptually similar to our work. However, we start from conversations and our structured data records are more complex.

6 Conclusion

In this paper we have presented an end to end method for extracting structured information from unstructured conversations using an encoder-decoder neural network. The restaurant-ordering domain we study is distinguished from past work by its conversational nature, and the need to handle user corrections and modifications. We incorporate a memory gate and role information into the encoder network to selectively keep important information and capture interaction patterns between conversation participants. Experimental results on both a human transcribed data set and ASR-recognized data set demonstrate the feasibility and effectiveness of our approach.

Acknowledgements

This work was done while Baolin Peng was an intern at Microsoft Research.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512, Cambridge, MA, October. Association for Computational Linguistics.

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

I. Beltagy and Chris Quirk. 2016. Improved semantic parsers for if-then statements. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 726–736, Berlin, Germany, August. Association for Computational Linguistics.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 4960–4964. IEEE.

David Chiang, Mona T. Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing arabic dialects. In Diana McCarthy and Shuly Wintner, editors, *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*. The Association for Computer Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of NIPS Deep Learning and Representation Learning Workshop*.

Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell. 2015. Language models for image captioning: The quirks and what works. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 100–105, Beijing, China, July. Association for Computational Linguistics.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany, August. Association for Computational Linguistics.

Hao Fang, Saurabh Gupta, Forrest N. Iandola, Rupesh Kumar Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. 2015. From captions to visual concepts and back. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1473–1482.

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

Chiori Hori, Takaaki Hori, Shinji Watanabe, and John R. Hershey. 2016. Context-sensitive and role-dependent spoken language understanding using bidirectional and attention lstms. In *Interspeech 2016*, pages 3236–3240.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems 27*, pages 2042–2050.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.

Rohit J. Kate and Raymond J. Mooney. 2006. Using string-kernels for learning semantic parsers. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 913–920, Sydney, Australia, July. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003*. The Association for Computational Linguistics.

Ioannis Konstas and Mirella Lapata. 2013. A global model for concept-to-text generation. *J. Artif. Intell. Res. (JAIR)*, 48:305–346.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany, August. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Hongyuan Mei, Mohit Bansal, and Matthew R. Walter. 2016. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 2772–2778.

Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Z. Hakkani-Tür, Xiaodong He, Larry P. Heck, Gökhan Tür, Dong Yu, and Geoffrey Zweig. 2015. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Trans. Audio, Speech & Language Processing*, 23(3):530–539.

Ramesh Nallapati, Bing Xiang, and Bowen Zhou. 2016. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1310–1318.

Chris Quirk, Raymond Mooney, and Michel Galley. 2015. Language to code: Learning semantic parsers for if-this-then-that recipes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 878–888, Beijing, China, July. Association for Computational Linguistics.

Gerard Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. 2013. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *CoRR*, abs/1312.6120.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2016. Context gates for neural machine translation. *CoRR*, abs/1608.06043.

Yuk Wah Wong and Raymond Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 439–446, New York City, USA, June. Association for Computational Linguistics.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Kaisheng Yao and Geoffrey Zweig. 2015. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 3330–3334. ISCA.

Kaisheng Yao, Geoffrey Zweig, and Baolin Peng. 2015. Attention with intention for a neural network conversation model. *CoRR*, abs/1510.08565.

Steve J. Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.