Scaling Multi-Domain Dialogue State Tracking via Query Reformulation

Pushpendre Rastogi
prastogi@amazon.com
Alexa AI
Amazon.com, Inc., USA

Arpit Gupta
arpgup@amazon.com
Alexa AI
Amazon.com, Inc., USA

Tongfei Chen *
tongfei@jhu.edu
Johns Hopkins University
Baltimore, MD,USA

Lambert Mathias
mathiasl@amazon.com
Alexa AI
Amazon.com, Inc., USA

Abstract

We present a novel approach to dialogue state tracking and referring expression resolution tasks. Successful contextual understanding of multi-turn spoken dialogues requires resolving referring expressions across turns and tracking the entities relevant to the conversation across turns. Tracking conversational state is particularly challenging in a multi-domain scenario when there exist multiple spoken language understanding (SLU) sub-systems, and each SLU sub-system operates on its domain-specific meaning representation. While previous approaches have addressed the disparate schema issue by learning candidate transformations of the meaning representation, in this paper, we instead model the reference resolution as a dialogue context-aware user query reformulation task —the dialog state is serialized to a sequence of natural language tokens representing the conversation. We develop our model for query reformulation using a pointer-generator network and a novel multi-task learning setup. In our experiments, we show a significant improvement in absolute F1 on an internal as well as a, soon to be released public corpora respectively.

1 Introduction

Dialogue *assistants* are used by millions of people today to fulfill a variety of tasks. Such assistants also serve as a digital marketplace[1] (Kumar et al., 2017) where any developer can build a domain-specific, task-oriented, dialogue *agent* offering a service such as booking cabs, ordering food, listening to music, shopping etc. Also, these agents may interact with each other, when completing a task on behalf of the user. Figure 1 shows one such interaction where the agent – ShopBot – must interpret the output of the agent – WikiBot. Often accomplishing this task requires understanding the context of a dialogue, communicating the conversational state to multiple agents and updating the state as the conversation proceeds.

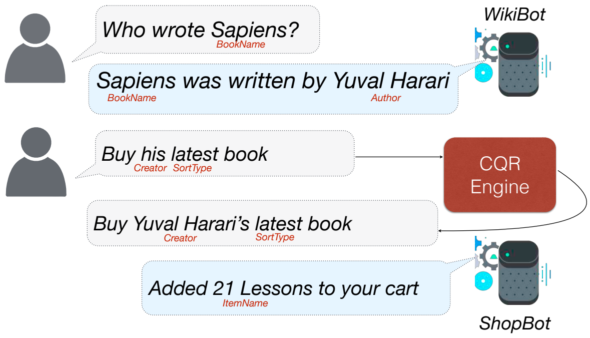

Figure 1: An example dialog where the second utterance by the user BUY HIS LATEST BOOK is reformulated as BUY YUVAL HARARI 'S LATEST BOOK. This reformulated user query is then input to SHOPBOT so that it can understand the user's request using its existing SLU logic for handling single-turn queries. This approach does not require any changes to the agent itself and can be scaled to multiple heterogeneous domains.

Tracking the dialogue state across multiple agents is challenging because agents are typically built for single-turn experiences, and must be laboriously updated to handle the context provided by other agents into their respective domain specific meaning representation. (Naik et al., 2018) proposed *context carryover*, a scalable approach to handle disparate schemas by learning mappings across the meaning representations, thereby eliminating the need to update the agents. However, the challenge of the agent's domain-specific SLU accuracy and choice of meaning representation remains. For example, in Figure 1 the SHOPBOT cannot handle pronominal anaphora and instead incorrectly labels HIS as the mention type CREATOR. Separately solving this problem for each agent, imposes

*Work done while the author was at Alexa AI
[1]https://dialogflow.com

Proceedings of NAACL-HLT 2019, pages 97–105
Minneapolis, Minnesota, June 2 - June 7, 2019.

a burden on the developer to relabel their data and update their SLU models, and is expensive and unscalable. Moreover, this approach cannot leverage the syntactic regularities imposed across agents by the natural language itself.

In this work, we propose a novel approach for enabling seamless interaction between agents developed by different developers by using natural language as the API. We build upon the pointer-generator network (PGN) proposed by (See et al., 2017) – originally for news article summarization – to rewrite user utterances and disambiguate them. Furthermore, we describe a new Multi-task Learning (MTL) objective to directly influence the attention of the PGN without requiring any extra manually annotated training data. Our results show that the new MTL objective reduces the error by 3.2% on slots coming from distances ≥ 3, compared to the basic PGN by (See et al., 2017).

2 Technical Details

Task We define a sequence of D dialogue turns, $\mathbf{x_t} = (u_{t-D+1}, r_{t-D+1}, \dots, u_{t-1}, r_{t-1}, u_t)$, where u_t is the user utterance at time t and r_t is the corresponding system response. $\mathbf{x_t}$ is the total information that our system has at time t. For example, the first row in Figure 2 shows $\mathbf{x_2}$ encoded as a single token sequence corresponding to the dialogue in Figure 1. The query rewriting task is to learn a function f_θ, with parameters θ, which maps $\mathbf{x_t}$ to its *rewrite* $\mathbf{y_t}$ which is another string, i.e. $\mathbf{y_t} = f_\theta(\mathbf{x_t})$. $\mathbf{y}_t$ should contain all the information needed by the agent to fulfill the user's request and it should be understandable by the agent as a standalone user request.

Model We use the pointer-generator (PGN) architecture (See et al., 2017) to construct f_θ. The PGN is a hybrid architecture which combines sequence-to-sequence model with pointer networks. This combination allows the PGN to summarize an input sequence by either *copying* from the input sentence, or *generating* a new word with a decoder RNN. We now describe the operation of the PGN in detail and focus on a single input sequence $\mathbf{x}$ with the subscript t omitted for simplicity. Let us slightly abuse notation and consider $\mathbf{x}, \mathbf{y}$ as sequences of tokens. We index the tokens of $\mathbf{x}, \mathbf{y}$ by l, k respectively. The PGN uses a two-layer Bi-Directional LSTM (BiLSTM) encoder to compute the hidden state vector h_l for $\mathbf{x}_l$.[2]

We now describe how y_k is generated. At time k, the probability of copying a token from the input p^{copy} is computed via a softmax over the attention weights computed using non-linear function of the encoder-LSTM hidden states $\mathbf{h}$ and the decoder LSTM's hidden state h_k^{decoder}. p^{mix} – a soft switch to decide between copying and generating – is computed using another non-linear function of h_k^{dec} and the final output distribution is given by

$$p(y_k) = p^{\text{mix}} p^{\text{gen}}(y_k) + (1 - p^{\text{mix}}) p^{\text{copy}}(y_k) \quad (1)$$

At decoding time, we can use either beam-search or greedily pick the token with the highest probability and move on to the next step. This is our baseline architecture for utterance rewriting.

Evaluation Ideally $\mathbf{y_t}$ should be judged as a correct rewrite if the downstream SLU system can parse $\mathbf{y_t}$, invoke the correct agent with the correct slots, and the agent can then take the right action. However, evaluating this notion of correctness would have required probing and instrumenting thousands of downstream agents and is not scalable to implement. Therefore, we used a simpler notion of correctness based on a manually collected set of *golden rewrites*, $\mathcal{Y}^*_{i,t}$, in this paper. Section 4.3 describes the metrics we use to evaluate our model's prediction $y_{i,t}$ against the golden set $\mathcal{Y}^*_{i,t}$.

Learning For training the model, we have a rewrites-corpus $\{\mathbf{x}_{it}, \mathbf{y}^*_{itj}\}_{i=1,t=1,j=1}^{I,T,J}$. I is the number of dialogs, T is the maximum number of turns in a dialog and J is the number of gold rewrites at a turn in a dialog. $\mathbf{y}^*_{i,t,j}$ denotes the j^{th} optimal rewrite for the user utterance at turn t in the i^{th} dialogue – $\mathbf{x}_{ti}$; $y^*_{i,t,j,k}$ is the k^{th} token in $\mathbf{y}^*_{\mathbf{i,t,j}}$. Our training objective is to maximize the log-likelihood:

$$\arg\max_\theta \sum_{i,t,j,k} \log p_\theta(y^*_{i,t,j,k}). \quad (2)$$

2.1 Multi Task Learning (MTL): Entity-Copy Auxiliary Objective

In Figure 2, both the references $y^*_{2,1}, y^*_{2,2}$ contain the same subset of entities – U3, and S1 – even though their order, and other tokens, in the gold rewrites have changed. This implies that for the task of rewriting utterances, the subset of entities that should be copied from the input dialog remains the same, irrespective of the dynamics of the decoder LSTM. Based on this observation we define

[2] For sake of brevity, we omit the update equations for the LSTM. Please refer to (See et al., 2017) for these details.

Input $x_{t=2}$	BOOKQUERY$_{l=1}$ Who wrote $\frac{\textit{Entity}\text{U1:BookName}}{\text{Sapiens}}_{l=4}$ $\frac{\text{SYSTEM}}{\text{INFORMINTENT}}$ $\frac{\textit{Entity}\text{U1:Sapiens}}{\text{Title}}$ was written by $\frac{\textit{Entity}\text{S1:Author}}{\text{Yuval Harari}}_{l=10}$ $\frac{\text{USER}}{\text{UNKINTENT}}$ Buy $\frac{\textit{Entity}\text{U2:Entity}}{\text{his}}$ $\frac{\textit{Entity}\text{U3:Entity}}{\text{latest}}$ book END$_{l=16}$
Refer. $\mathcal{Y}^*_{t=2}$	$\{y^*_{2,j=1}=$ Buy$_{k=1}$ *Entity*S1 *Entity*U3 book$_{k=4}$, $y^*_{2,2}=$ Buy *Entity*U3 book by *Entity*S1 $\}$

Figure 2: An example of sequential input received by our utterance disambiguation seq2seq model and a list of reference outputs. The words in short-caps denote the domain and intent predicted by the SLU system which are concatenated to the beginning of the sequence. Words beginning with *Entity* are placeholders used to delexicalize names of entities. Both references 1 and 2 are input to the SLU system during training. We explicitly named the indices at a few locations to aid the reader.

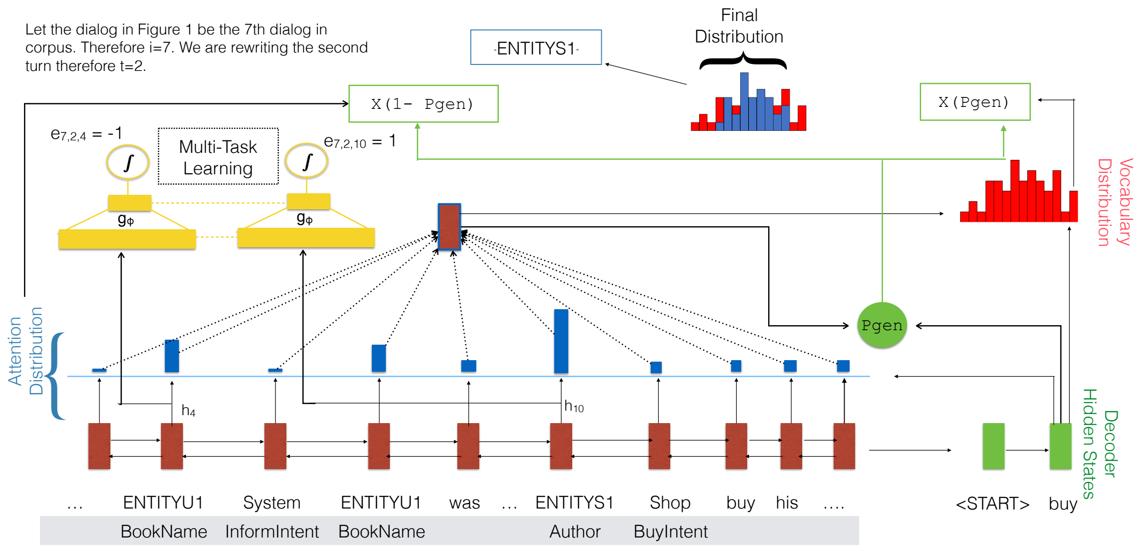

Figure 3: Model Architecture of the CQR Model which performs Multi-Task Learning for Pointer-Generator Networks. We show a snapshot just before decoder generates the word ENTITYS1 or *Yuval Harari*. Also, we show for MTL ENTITYU1 gets the label -1 as it is not one of the final slots, and ENTITYS1 gets a label of 1

an auxiliary task and augment the learning objective as shown in Figure 3.

As mentioned earlier, the copy distribution p_k^{copy} is a function of the encoder hidden state $\mathbf{h} = (h_1, \ldots, h_l, \ldots, h_{|x|})$ which does not change with k. If x_l was an entity token then h_l should be informative enough to decide whether that token should be copied or not. Therefore, we add a two layer feed-forward neural network, g_ϕ, that takes h_l as input and predicts whether the l^{th} token should be copied or not. Given the probability $g_\phi(h_l)$ we minimize the binary cross-entropy loss, and back-propagate through h_l which influences θ. The auxiliary objective should improve the generalization because it forces the encoders representation to become more informative about whether an entity should be copied or not. At inference time g_ϕ is not used. Formally, let $e_{i,t,l}$ take the following value:

$$e_{i,t,l} = \begin{cases} 1 & \text{if } x_{i,t,l} \text{ is an entity and } x_{i,t,l} \in \mathcal{Y}^*_{i,t} \\ -1 & \text{if } x_{i,t,l} \text{ is an entity and } x_{i,t,l} \notin \mathcal{Y}^*_{i,t} \\ 0 & \text{Otherwise} \end{cases}$$

Let $\lambda > 0$ be a hyperparameter. We add a binary log-likelihood objective to objective 2 to create objective 3. We refer to the PGN model trained with objective 3 as **CQR** in Table 4.

$$\sum_{i,t,j,k} \log p(y^*_{i,t,j,k}) + \lambda \sum_{i,t} \sum_{l=1}^{|x_{i,t}|} e_{i,t,l} \log g_\phi(h_{i,t,l}) \quad (3)$$

3 DataSet and Preprocessing

In this section we will describe how we created the golden rewrites $\{\mathcal{Y}^*_t \mid \forall t\}$ for each of the above datasets and our pre-processing steps that we found crucial to our success.

3.1 Generating gold rewrites

We used two separate approaches to generate gold rewrites for the INTERNAL and INCAR datasets. For the INCAR dataset we collected 6 rewrites for each utterance that had a reference to a previously mentioned entity.[3] For the INTERNAL dataset, which has over 100K sentences the above approach would be prohibitively expensive. Therefore, instead of gathering completely new manual annotation we used a semi-manual process. We utilized a template dataset that is keyed by the *Domain, Intent* and *Slots* present in that utterance and contains the top-5 most common and unambiguous phrasing for that key. For example to create the rewrite in Figure 1 we filled the template:

Buy Creator 's SortType ItemType

This template was chosen randomly from other valid alternatives such as Buy SortType ItemType by Creator . These valid alternatives were determined on the basis of existing manual *domain, intent,* and *schema* SLU annotations which indicated which slots were required to answer the user's utterance.

3.2 Role-based Entity Indexing

In this step, the entity words in x_t are replaced with their canonical versions. Our results show that this significantly improved both BLEU and Entity F1 measures. To replace entity words we use string matching methods to extract tokens for dialogue. We maintain two separate namespaces for user entities and system entities respectively. However, if an entity appears again in dialogue, we do ***not*** assign it a new canonical token but used already assigned one. Also, as seen in Figure 2 we also add the entity tag to slot representation. Lastly, as re-writing happens before any SLU component we do not have this information for u_t. In u_t we only replace entities with canonical tokens, but do not add any information about entity. Table 1 show how to transform dialogue from Figure 1.

[3]https://github.com/alexa/alexa-dataset-contextual-query-rewrite

Before	After Pre-Processing
Who wrote Sapiens	who wrote U_1||BookName
Sapiens was written by Yuval Harari	U_1||Author was written by S_1||BookName
Buy his most recent book	Buy U_3||UNK U_4||UNK book

Table 1: Replacing entities with the role-based canonical versions.

3.3 Abstractified Possessives

Generalizing on rare words and rare contexts is the true test of any NLP system, and linguists have long argued in favor of syntactically motivated models that abstract away from lexical entries as much as possible (Klein and Manning, 2003). In this preprocessing step, we show the benefit of such abstraction. While testing the PGN architecture we noticed that the sequence decoder would sometimes generate an off-topic rewrite if the input sequence contained a rare word. In order to avoid this problem we augmented the input sequence with additional features to mark the syntactic function of words. Specifically we used the Google Syntactic N-gram Corpus (Goldberg and Orwant, 2013) to add syntactic features to each word in the dialogue. We harvested a list of top 1000 words that appear most frequently after possessive pronouns. We concatenated three types of extra features to the words in a dialogue. The first feature was the QUESTION feature which was concatenated to the 7 question words. The second feature was the $PRP\$$ tag which we concatenated to specific possessive pronouns. Finally we added a tag called PSBL – short for possessible – for the top 1000 words that we found from the Syntactic N-Gram Corpus.

We decided not to use POS tags because we did not have manually POS tagged data on our domain and off-the-shelf POS tagger[4] did not perform well on our dataset.

4 Experiments

4.1 Dataset

We used two datasets to evaluate our method. The first is a public dataset (Regan et al., 2019) we call INCAR, which is an extension to (Eric and Manning, 2017). The dataset consists of $3,031$ dialogues from three domains: Calendar Scheduling, Weather, and Navigation, that are useful for an in-car conversational assistant. We crowd-sourced six

[4]https://spacy.io/

rewrites for each utterance in the corpus that had a reference to previously mentioned entities. The second dataset, called INTERNAL, is an internal benchmark dataset we collected over six domains – weather, music, video, local business search, movie showtimes and general question answering. Table 2 describes the data statistics for this internal collection. About 40% of the dialogues in this corpus are cross-domain, which makes it much harder than the INCAR dataset.

Context Length	Train	Dev	Test
1	125K	42K	21K
2	8K	3k	1K
>=3	4K	1K	700

Table 2: INTERNAL data statistics. Each turn consists of a user and a system turn i.e context length = 2 implies two turns.

4.2 Training

We used OpenNMT (Klein et al., 2017) toolkit for all our experiments. We modified it to include the multi-task loss function as described in Section 2.1. Unless explicitly mentioned here, we used the default parameters defined in OpenNMT recipe. Various hyper-parameters were tuned on a reduced training set and the development set. Our encoder was a 128-dimensional bi-directional LSTM. We used the Adagrad optimizer with a learning rate of 0.15, and we randomly initialized 128-dimensional word embeddings. The word embeddings were shared between the encoder LSTM and the decoder LSTM. λ in Eq.3 was set to 0.01. We trained the model for 20 epochs with early stopping on a validation set.

4.3 Evaluation Metrics

BLEU: has been widely used in machine translation tasks (Papineni et al., 2002), dialogue tasks (Eric and Manning, 2017), and chatbots (Ritter et al., 2011). It gives us an intrinsic measure to evaluate quality of re-writes without caring about downstream SLU evaluation.

Response Entity F1 (ResF1): We measure this metric for the INCAR dataset, following the approach outlined by (Madotto et al., 2018)[5]. The Response Entity F1 micro-averages over the entire set of system responses and compare the entities in plain text. The entities in each gold system response are selected by a predefined entity list. This metric evaluates the ability to generate relevant entities and to capture the semantics of the dialogue. We reimplemented the *Mem2SeqH1* architecture in (Madotto et al., 2018)[6] and we refer to our implementation as *Mem2Seq**. We use utterances produced by our proposed **(CQR)** system in the dialogue instead of original utterances while evaluating using Mem2Seq*. Note that our reimplementation, Mem2Seq*, achieves a Response Entity F1 of 33.6 which is higher than the best overall Entity F1 score of 33.4 reported in (Madotto et al., 2018).

Entity F1: This measures micro F1 between entities in the hypothesized rewrite and gold rewrite. This is different from F1 reported by (Madotto et al., 2018) as they evaluate F1 over system entities, whereas here we evaluate the entities over the user turn. We employ a recent state-of-art bi-directional LSTM with CRF decoding (Ma and Hovy, 2016) to implement our SLU system.

5 Results

5.1 INTERNAL Dataset Results

On INTERNAL dataset we show CQR significantly improves over (Naik et al., 2018) in Table 4. **CQR** also improves F1 for current turn slots as it can leverage context and distill necessary information to improve SLU. Further, we can see that most improvements upon the baseline PGN model (M0) come from pre-processing steps like canonicalizing entities. In the baseline model, it has to learn to generate entity tokens individually, whereas in **M1** the model only has to learn to copy tokens like *USER_ENT_1*. Finally, our proposed multi-task learning model (CQR) improves both BLEU and EntityF1 at most distances. Specifically, we see an improvement of 4.2% over **M2** for slots at distances $\geq$3. In Table 4 distance is measured differently from Table 2, here we count User and System turns individually to showcase how distance affects **EntityF1**. If an entity is repeated multiple times in the context, we consider its closest occurrence to report results.

5.2 INCAR Dataset Results

For INCAR dataset we pick the best model **CQR** from Table 4 and re-train on the respective dataset. On the navigation domain we observe significant

[5]Evaluation script available at https://github.com/HLTCHKUST/Mem2Seq

[6]The *Mem2Seq H1* was the best performing system in terms of ResF1, in two out of three domains in the InCar dataset, and it was the fastest Mem2Seq model. Therefore, we used *Mem2SeqH1* and not *Mem2SeqH3*

Dialogue	U: Find me a Starbucks in Redmond S: I found a Starbucks in Redmond WA. It's 15.7 miles away on NE 76th St. It's open now until 9:00 PM. U: How do I get there?	
PGN	how can i get to redmond	
CQR	how do i get to the starbucks on NE 76th St WA	
Gold	how do i get to the starbucks on NE 76th St	
Dialogue	U: How is the weather tomorrow? S: In Chicago there will be mostly sunny weather U: What about saturday?	
PGN	what is the weather in chicago on saturday ?	
CQR	what is the weather in chicago on saturday ?	
Gold	what is the weather in chicago on saturday ?	

Table 3: Examples of generated responses for Internal Dataset

improvement. We believe this is because there are on average 2.3 slots were referred from history in rewrites requiring copy from dialog as compared to 1.3 and 1.1 in schedule and weather domain respectively. Also, we compare with an oracle CQR (i.e., gold-rewrite from our data collection, instead of predicted re-write) to measure the potential of query-rewriting and motivate further research on this topic. We can see that the CQR model performs better than the Mem2Seq* model, indicating that query rewriting is a viable alternative to dialogue state tracking. This is important in environments where changing the NLU systems to leverage memory structures is not always feasible. We claim that query rewriting is a simpler approach in such situations, with no loss in performance.

6 Related Work

Probabilistic methods for task-oriented dialogue systems typically divide an automatic dialogue agent into modules such as automatic speech recognition(ASR) for converting speech to text, spoken language understanding(SLU) for classifying the domain and intent of the current utterance and tagging the slots in the current utterance, dialogue state tracking(DST) for tracking what has happened in the dialogue so far, and dialogue policy for deciding what actions to take next (Young, 2000). In this traditional framework, SLU is seen as a low-level task that interprets the user's current utterance in isolation, without accounting for the dialogue history. For example, in Figure 1 the platform system parses the utterance WHO WROTE SAPIENS, to infer that the user intends to query for information about a book, and then the platform performs BIO style tagging with an intent-specific schema to label the mentionSAPIENS as the slot key BOOKNAME. Most SLU systems perform this without any context information. Some recent work focussed on contextual SLU (Shi et al., 2015; Liu et al., 2015; Chen et al., 2016) propose memory architectures to incorporate contextual information while performing the SLU step. However because their task was restricted to domain-intent classification and slot tagging for the current utterance only, a higher level DST module is still required to combine information from previous turns with the current utterance to create a single dialogue state.

DST is considered to be a higher-level module as it has to combine information from previous user utterances and system responses with the current utterance to infer its full meaning. Many deep-learning based methods have recently been proposed for DST such as neural belief tracker (Mrkšić et al., 2017), and self-attentive dialogue state tracker (Zhong et al., 2018) which are suitable for small-scale domain-specific dialogue systems; as well as more scalable approaches such as (Rastogi et al., 2017; Xu and Hu, 2018) that solve the problem of an infinite number of slot values and (Naik et al., 2018) who additionally solve the problem of huge number of disparate schemas in each domain. End-to-End approaches based on deep learning have also been proposed recently to replace such modular architectures, like (Madotto et al., 2018; Eric and Manning, 2017).

Unfortunately, all of the above approaches fail to address the problem that, as the number of domain-specific chatbots on a dialogue platform grows larger, the DST module becomes increasingly complex as it tries to handle the interactions between different chatbots and their different schemas. For example, consider the scenario shown in Figure 1. Chatbot A, the BOOK chatbot, can understand domain-specific utterances like "who wrote X ?" annotated with a special schema with slot keys such as BOOKNAME, AUTHOR. In order to disambiguate utterance u_2 the DST in the conversational platform must know that the CREATOR slot-key in the SHOPPING chatbot co-refers to the AUTHOR slot-key. However, this leads to a quadratic explosion in the number of possible transitions that the platform has to learn, thereby significantly increasing the learning problem for DST. Additionally, the problem is more challenging than just disambiguating pronouns because in some situations there may

System	Entity F1												BLEU
	d=0			d=1			d=2			d≥3			
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	
(Naik et al., 2018)	95.1	**95.1**	95.1	74.9	78.4	76.6	72.2	82.2	76.9	10.4	46.3	17.0	N/A
PGN (M0)	**99.0**	78.1	87.4	**95.9**	62.1	75.4	**95.2**	57.3	71.5	87.3	65.5	74.9	83.4
+Canonical Ent. (M1)	98.7	93.9	**96.3**	92.9	93.5	93.2	94.4	96.9	95.6	69.8	78.5	73.9	89.9
+Syntax Info (M2)	98.6	93.9	96.2	92.9	93.5	93.2	94.3	96.9	95.6	69.8	78.5	73.9	89.9
+MTL (CQR)	98.5	**94.0**	96.2	93.7	**93.8**	**93.7**	94.2	**97.4**	**95.8**	**75.2**	**79.0**	**77.1**	**90.3**
% Relative Improv.	3.6	-1.2	1.2	25.1	19.6	22.3	30.5	18.5	24.6	623.1	70.6	353.5	N/A

Table 4: Comparison of Pointer-Generator variants to traditional state tracking approach on the INTERNAL dataset. We measure entity F1 across slots from different distances separately. Slot distance is counted per utterance starting from the current user utterance. Therefore, slots at d=0 are slots from the current user utterance that should have been copied. d=1 refers to slots from system response in the last turn, d=2 refers to slots from the user in last turn and d≥3 aggregates all other turns. $d \geq 3$ is the most challenging test-subset where **CQR** has the highest benefit.

System	E2E BLEU	ResF1 All	Schedule	Weather	Navigation
Mem2Seq*	11.4	33.6	**48.4**	47.2	19.4
CQR	**11.6**	**36.1**	**48.4**	**47.9**	**23.8**
% Relative Improv.	1.8	7.4	0.0	1.5	22.7
CQR-Oracle	11.8	38.0	48.9	48.9	26.9

Table 5: Comparison of PGN variants proposed in this paper on the INCAR dataset in comparison to the state tracking approach. Our proposed CQR model outperforms the MemSeq* system, which is a stronger baseline than the Mem2Seq results published in Madotto et al. (2018).

be no co-referent pronouns in the current utterance. For example, a user may say "what's the address" instead of saying "what is its address", creating a case of zero-anaphora.

Finally, we will mention that Seq2Seq models with Attention (Sutskever et al., 2014; Bahdanau et al., 2014) have seen rapid adoption in automatic summarisation (See et al., 2017; Rush et al., 2015). Exploring black-box methods like query re-writing allow us to benefit from the progress made in these fields and apply them to state tracking and reference resolution tasks in dialogue.

7 Conclusion

In this work we made three fundamental contributions. First, we proposed *contextual query rewriting*(CQR) as a novel way to interpret an input utterance in context given a dialogue history. For example, we can rewrite BUY HIS LATEST BOOK as BUY YUVAL HARARI'S MOST RECENT BOOK, given the dialogue history, as shown in Figure 1. The output of CQR can directly be fed to the domain-specific downstream SLU system which drastically simplifies the construction of task-specific dialogue *agents*. Since we do not need to change either the spoken language understanding or the dialogue state tracker downstream, our approach is a black-box approach that improves the modularity of industrial-scale, dialogue-asistants. Second, we investigated how to optimally use a Pointer-Generator Network for the CWR task using Multi-Task Learning and task-specific preprocessing. Finally, we demonstrated the efficacy of our approach on two datasets. On INCAR dataset released by (Eric and Manning, 2017), we were able to show that re-writing of the user utterance can benefit end-to-end models. On a proprietary INTERNAL dataset we showed that our approach can greatly improve the experience when referring to entities from much further away in a dialogue history, resulting in relative improvements in Entity F1 of greater than 20% on the most challenging subset of the test-data.

We hope that our approach of directly using natural language as an api will motivate other researchers to conduct work in this direction.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yun-Nung Vivian Chen, Dilek Hakkani-Tr, Gokhan Tur, Jianfeng Gao, and Li Deng. 2016. End-to-end memory networks with knowledge carryover for multi-turn spoken language understanding. In *17th Annual Meeting of the International Speech Communication Association*. ISCA.

Mihail Eric and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. *arXiv preprint arXiv:1705.05414*.

Yoav Goldberg and Jon Orwant. 2013. A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, volume 1, pages 241–247.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Anjishnu Kumar, Arpit Gupta, Julian Chan, Sam Tucker, Bjorn Hoffmeister, Markus Dreyer, Stanislav Peshterliev, Ankur Gandhe, Denis Filiminov, Ariya Rastrow, et al. 2017. Just ask: building an architecture for extensible self-service spoken language understanding. *arXiv preprint arXiv:1711.00549*.

Chunxi Liu, Puyang Xu, and Ruhi Sarikaya. 2015. Deep contextual language understanding in spoken dialogue systems. In *Sixteenth annual conference of the international speech communication association*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1064–1074.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. *arXiv preprint arXiv:1804.08217*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Chetan Naik, Arpit Gupta, Hancheng Ge, Lambert Mathias, and Ruhi Sarikaya. 2018. Contextual slot carryover for disparate schemas. In *19th Annual Meeting of the International Speech Communication Association*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Abhinav Rastogi, Dilek Hakkani-Tür, and Larry Heck. 2017. Scalable multi-domain dialogue state tracking. In *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*, pages 561–568. IEEE.

Michael Regan, Pushpendre Rastogi, Arpit Gupta Gupta, and Lambert Mathias. 2019. A dataset for resolving referring expressions in spoken dialogue via contextual query rewrites (cqr). *arXiv preprint arXiv:1903.11783*.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing*, pages 583–593. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics.

Yangyang Shi, Kaisheng Yao, Hu Chen, Yi-Cheng Pan, Mei-Yuh Hwang, and Baolin Peng. 2015. Contextual spoken language understanding using recurrent neural networks. In *ICASSP 2015*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NIPS*, page 9.

Puyang Xu and Qi Hu. 2018. An end-to-end approach for handling unknown slot values in dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1457, Melbourne, Australia. Association for Computational Linguistics.

Steve J Young. 2000. Probabilistic methods in spoken–dialogue systems. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 358(1769):1389–1402.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, Melbourne, Australia. Association for Computational Linguistics.