Explicit State Tracking with Semi-Supervision for Neural Dialogue Generation

Xisen Jin[§*], Wenqiang Lei[†*], Zhaochun Ren[‡], Hongshen Chen[‡], Shangsong Liang[*],
Yihong Zhao[‡], Dawei Yin[‡]
‡JD.com, Beijing, China
§Fudan University, Shanghai, China
†National University of Singapore, Singapore
*King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
xisenjin@gmail.com,wenqiang@comp.nus.edu.sg,renzhaochun@jd.com
chenhongshen@jd.com,shangsong.liang@kaust.edu.sa,ericzhao@jd.com,yindawei@acm.org

ABSTRACT

The task of dialogue generation aims to automatically provide responses given previous utterances. Tracking dialogue states is an important ingredient in dialogue generation for estimating users' intention. However, the *expensive nature of state labeling* and the *weak interpretability* make the dialogue state tracking a challenging problem for both task-oriented and non-task-oriented dialogue generation: For generating responses in task-oriented dialogues, state tracking is usually learned from manually annotated corpora, where the human annotation is expensive for training; for generating responses in non-task-oriented dialogues, most of existing work neglects the explicit state tracking due to the unlimited number of dialogue states.

In this paper, we propose the *semi-supervised explicit dialogue state tracker* (SEDST) for neural dialogue generation. To this end, our approach has two core ingredients: *CopyFlowNet* and *posterior regularization*. Specifically, we propose an encoder-decoder architecture, named *CopyFlowNet*, to represent an explicit dialogue state with a probabilistic distribution over the vocabulary space. To optimize the training procedure, we apply a posterior regularization strategy to integrate indirect supervision. Extensive experiments conducted on both task-oriented and non-task-oriented dialogue corpora demonstrate the effectiveness of our proposed model. Moreover, we find that our proposed semi-supervised dialogue state tracker achieves a comparable performance as state-of-the-art supervised learning baselines in state tracking procedure.

KEYWORDS

Dialogue generation, Dialogue state tracking, Semi-supervised learning, Posterior regularization

1 INTRODUCTION

In recent years, dialogue systems have received increasing attention in numerous web applications [1, 3, 36, 58]. Existing dialogue systems can fall into two categories: non-task-oriented dialogue systems and task-oriented dialogue systems. Non-task-oriented dialogue systems aim to generate fluent and engaging responses, whereas task-oriented dialogue systems need to complete a specific task, e.g., restaurant reservation, along with a response generation process. Employing neural networks to generate natural and sound responses, the task of *neural dialogue generation* is playing an important role in dialogue systems [5, 23, 24, 31, 39, 41, 44, 45, 50]. In a dialogue, a *dialogue state* refers to a full and temporal representation of each participant's intention [12]. Thus in neural dialogue generation, dynamically tracking dialogue states is the key for generating coherent and context-sensitive responses.

Numerous dialog state tracking mechanisms with a limited state space have been proposed for task-oriented dialogue systems, e.g., hand-crafted rules [12, 47], conditional random fields [18, 19, 34], maximum entropy [51], and neural networks [15]. As a state-of-the-art work, *explicit dialog state tracking via an interpretable text span* has been preliminarily attempted on task-oriented dialogue systems [20]. Differently, in non-task-oriented dialogue systems, most of existing state tracking approaches employ a fixed-size latent vector to represent the whole dialogue history [39]. Though these solutions are capable for chit-chat conversations, they fail to distinguish similar concepts or entities(e.g., product names) which are often key information in technical and transactional domains [5]. Moreover, these latent vectors have weak interpretability. However, existing solutions to explicit state tracking cannot be applied in non-task-oriented dialogue systems, since these supervised approaches typically require large amounts of manually annotated dialogue states.

---

* Work performed during an internship at JD.com.

Accordingly, the neural dialogue generation faces a dilemma between reducing the expense of data annotation and improving the performance of dialogue state tracking: (1) For both task-oriented and non-task-oriented dialogue generation, existing explicit approaches requires a large amounts of manually labeled data to train the state tracker [20]. Heavily relying on the expensive annotated corpus, these methods lead the state tracker extremely difficult to be transferred to new scenarios or extended to a larger state space. (2) Most of unexplainable state trackers in non-task-oriented dialogue generation are not capable of explicitly tracking long-term dialogue states, limiting their capability in complicated domains. To tackle the above challenges, our focus is on developing methods to construct an explicit dialogue state tracker with unlabeled data for neural dialogue generation.

In this paper, we propose a semi-supervised neural network, named the *semi-supervised explicit dialogue state tracker* (**SEDST** for short), to explicitly track dialogue states for both task-oriented and non-task oriented dialogue generation with a text span. Along with SEDST, we propose a novel encoder-decoder architecture based on copying mechanism [14], called *CopyFlowNet*, to represent dialogue states with explicit word sequences. We infer these word sequences, i.e., text spans, through a probabilistic distribution over the vocabulary space. To optimize the training procedure of SEDST, we employ a posterior regularization strategy to integrate indirect supervision from unlabeled data. Thus SEDST is compatible for both supervised and unsupervised learning scenarios.

In our experiments, we verify the effectiveness of our proposed method in both task-oriented dialogue generation and non-task-oriented dialog generation, respectively. We find that SEDST, under 50% data annotated setup, outperforms state-of-the-art task-oriented dialogue generation baselines, as well as outperforms non-task-oriented baselines under no data annotated setup. Moreover, we deeply analyze and verify the effectiveness of the posterior regularization strategy in incorporating indirect supervision for the dialogue state tracking.

To sum up, our main contributions can be summarized as follows:

- We focus on tracking explicit dialogue states with semi-supervision for neural dialogue generation.
- We propose a semi-supervised neural dialogue generation framework, called **SEDST**, for both task-oriented and non-task-oriented dialogue systems.
- We propose an explicit dialogue state tracker, CopyFlowNet, with implicit copyNets and posterior regularization.
- We verify the effectiveness of SEDST in our extensive experiments on both task-oriented copora and non-task-oriented copora.
- We deeply study and analyze the performance of SEDST and other widely used two-stage decoding models in dialogue systems.

We introduce related work in §2. We provide preliminaries in §3 and describe our approach in §4. Then, §5 details our experimental setup, §6 presents the results, and §7 concludes the paper.

2 RELATED WORK

We detail our related work on two lines: neural dialogue generation and dialogue state tracking.

2.1 Neural dialogue generation

Neural dialogue generation aims at generating natural-sounding replies automatically to exchange information, e.g., knowledge [7, 42, 58]. As a core component of both task-oriented and non-task-oriented dialogue systems, neural dialogue generation has received a lot attention in recent years [1, 3, 36, 58]. Among all these approaches, sequence-to-sequence structure neural generation models [6, 7, 23, 24, 39, 41, 44, 45] have been proved to be capable in multiple dialogue systems with promising performance. Several approaches have been proposed to softly model language patterns such as word alignment and repeating into sequence-to-sequence structure [2, 6, 14, 40, 56]. Xing et al. [56] employ attention mechanism [2] to dynamically incorporates contextual information into response generation. Gu et al. [14] propose a copy mechanism to consider additional copying probabilities for contextual words in forum conversations. Serban et al. [40] decodes coarse tokens before generating the fulls response. Cao and Clark [6] tackle the boring output issue of deterministic dialogue models by introducing a latent variable model for one-shot dialogue response. In [46], the authors selects and predicts explicits keywords using before response generation. Zhang et al. [59] use a log-linear model to represent the desired distribution and inject the prior knowledge by a posterior regularization. Additionally, recent work verify that reinforcement learning is a promising paradigm when state and action spaces are carefully designed [9, 55].

2.2 Dialogue state tracking

Dialogue state tracking is an important ingredient of the dialogue generation. Traditional methods utilize hand-crafted rules to select the dialogue state [12]. Relying on the most likely results from an natural language understanding (NLU) module [32], these rule-based systems hardly models uncertainty, which is prone to frequent errors [32, 54]. Young et al. [57] propose a distributional dialogue state for statistical dialog system and maintain a distribution over multiple hypotheses facing with noisy conditions and ambiguity. Another typical form of dialogue state is in the form of a probability distribution over each slot for each turn [52, 53].

In task-oriented dialogue systems, end-to-end neural networks have been successfully employed for tracking dialogue states via interacting with an external knowledge base [5, 10, 50, 55]. Wen et al. [50] divide the training procedure into two phases: the dialogue state tracker training, and the whole model training. Mrkšić et al. [29] proposed a dialogue state tracker based on word embedding similarities. Eric and Manning [10] implicitly model a dialogue state through an attention-based retrieval mechanism to reason over a key-value representation of the underlying knowledge base. Bordes et al. [5] memories the dialogue context in a memory module and repeatedly queries and reasons about this context to select an adequate system response. Instead of employing symbolic knowledge queries, Dhingra et al. [9] propose an induced "soft" posterior distribution over the knowledge base to search matching entities. Lei et al. [20] proposed an extendable framework to track dialogue states with a text span including the constraints for a knowledge base query.

In non-task-oriented dialogue systems such as forum conversations, lots of efforts have been made to keep track of the dialogue

Table 1: Glossary.

Symbol	Description
D	a dialogue session
U	a user utterance
R	a machine response
S	a text span for dialogue state tracking
X	an input sequence
Y	an output sequence
s	an element in a text span
x	an element in a source sequence
y	an element in a target sequence
N	length of an utterance, text span and a machine response
w_x	a word input in source sequence
$\mathbf{h}$	a hidden vector generated in a GRU.
e	an entity obtained in the knowledge base search
$\mathcal{I}$	a user's intention in a dialogue
$\mathcal{V}$	vocabulary
$\mathcal{K}$	a knowledge base
$\mathcal{A}$	a set of annotated training instances
$\mathcal{U}$	a set of unannotated training instances

process in multi-turn settings [38, 39, 43]. Serban et al. [39] and Sordoni et al. [43] incorporates hierarchical structures in word and sentence levels to encourage cohesive multi-turn dialogue generation. Serban et al. [38] utilize a latent variable at the sub-sequence level in a hierarchical setting. Chen et al. [7] add a hierarchical structure and a variational memory module into a neural encoder-decoder network. However, all these latent vectors and latent memories are unexplainable, which makes it challenging to verify the effectiveness of dialogue state tracking. Moreover, these unexplainable latent vectors fail to distinguish distinctive concepts with similar vector representation, e.g., product names.

Our work differs from previous work in the following important ways: (1) We represent dialogue states in text spans explicitly; (2) We propose the CopyFlowNet which enables semi-supervised and unsupervised training of the state tracker; (3) We propose a novel training method incorporating posterior regularization to improve the robustness.

3 PRELIMINARIES

In this section, we first formalize task-oriented and non-task-oriented dialogue systems. Thereafter, we introduce the approach of tracking dialogue state using text spans.

3.1 Problem formulation

Before introducing our method for dialogue state tracking, we introduce our notation and key concepts. Table 1 lists our notations in this paper. Given T dialogue turns, a *dialogue session D* consists of a sequence of utterances, i.e., $D = \{U_1, R_1, U_2, R_2.., U_T, R_T\}$, where U, R refers to responses from a user and a machine respectively. At the t-th turn, given the current user utterance U_t and historical records $U_1, R_1, U_2, R_2.., U_{t-1}, R_{t-1}$, the dialogue system generates a response R_t. Probabilistically, the system generates R_t via maximizing the probability $P(R_t|U_1, R_1, U_2, R_2.., U_{t-1}, R_{t-1})$, shown in Eq. 1:

$$R_t = \arg\max P(R_t|U_1, R_1, U_2, R_2.., U_{t-1}, R_{t-1}, U_t), \quad (1)$$

Task-oriented dialogue system includes the *task completion* component which is specified by users (e.g., reserving an restaurant) [49]. To further formulate the task completion, we denote the user intention as $\mathcal{I}$. In this paper, we simplify the user's intention as a specific entity e, searched in knowledge bases at the end of a dialogue. *Non-task oriented systems*, differently, just focus on generating engaging and coherent responses. Typically they involve broader domains and more complicated contextual information [40, 46]. Note that it is difficult to find a clear boundary between task oriented dialogue systems and non-task oriented dialogue systems. To clarify the difference between these two types of dialogue systems, in this paper any dialogue system including knowledge base interactions refers to a task-oriented dialogue system.

Dialogue state tracking is the key component for both non-task-oriented dialogue systems and task oriented dialogue systems. In non-task oriented dialogue systems, dialogue state tracking is a key to generate context-aware and coherent responses. Whereas in task-oriented dialogue systems, dialogue state tracking is becoming mandatory since it has to capture users' request and constraints for knowledge base search.

3.2 Dialogue state tracking using text spans

Text-span based dialogue state trackers have been proposed to manifest simplicity along with better interpretability [20, 49]. At t-th turn, such state tracker employs a text span S_t (i.e., a sequence of words) to track dialogue states. S_t aims at summarizing past utterances and responses (i.e., $U_1, R_1, U_2, R_2,...,U_{t-1}, R_{t-1}, U_t$).

We follow the notation from the Sequicity [20], which is a state-of-the-art dialogue state tracker for the task-oriented dialogue generation. Sequicity defines a text span over the full vocabulary space called *bspan* (denoted as S_t), which records all "requestable slots" and "informable slots" [49], each separated with delimiters. Informable slots track the constraints which are used for knowledge base search while requestable slots record what users are looking for in current dialogues. With bspan, knowledge base search can be performed by taking informable slots as search constraints. As such, task completion can be converted as a problem of generating a text span S_t at each turn.

Generating a text span S_t of keywords also improves performance of single response generation [40, 46]. We elaborate this strategy into dialogue state tracking within multi-turn dialogues for non-task oriented dialogues. In our work, we define a *state span* as a text span that indicates a dialogue state. Shown in Eq. 2, in this paper the problem of both task-oriented and non-task-oriented dialogue generation can be decomposed into two successive steps: (1) generating a state span S_t; (2) generating the response R_t.

$$\begin{aligned} S_t &= \arg\max P(S_t|U_1, R_1, U_2, R_2.., U_{t-1}, R_{t-1}, U_t), \\ R_t &= \arg\max P(R_t|S_t, U_1, R_1, U_2, R_2.., U_{t-1}, R_{t-1}, U_t). \end{aligned} \quad (2)$$

4 METHOD

In this section, we propose our semi-supervised explicit dialogue state tracker, abbreviated as **SEDST**. We start by providing an

Figure 1: Overview of our dialogue model with semi-supervised explicit dialogue state tracker. (A) provides an overview of dialogue response generation process. Each arrow represents a *copyflow*(see 4.2). (B) provides details of the t-th dialogue turn. (C) represents the knowledge base interaction for task-oriented dialogues. "Parking garage" is the constraint for knowledge base search in the example. Ellipses are the input and output tokens and rectangles denote neural hidden vectors.

overview of SEDST. We then describe the CopyFlowNet architecture and detail our posterior regularization for optimizing the model training.

4.1 Overview

We propose the *semi-supervised explicit state tracker* (**SEDST**) to track dialogue states with explicit text spans under no or few data annotation setup. SEDST includes two main ingredients: (1) Copy-FlowNet; (2) posterior regularization. Along with SEDST, we propose an encoder-decoder architecture, *CopyFlowNet*, to generate explicit state spans by copying from existing sequences on top of normal generation process. Finally, we provide an optimized training procedure of SEDST, where we apply a posterior regularization strategy to improve the robustness of our model.

Figure 1 provides an overview for state tracking and response generation process in SEDST. The turn-level overview presents how the text span is utilized for state tracking for multi-turn dialogues, where each arrow represents a *copyflow*(see 4.2). The sentence level illustration presents encoding and decoding procedure. We also illustrate how the knowledge-base interaction is performed with state spans for task-oriented dialogues, where we follow [20].

4.2 CopyFlowNet

In this section we detail the *CopyFlowNet* architecture. At t-th dialogue turn, for simplicity we set the length of U_t and R_{t-1} to N; CopyFlowNet first encodes the concatenation of previous response R_{t-1} and current user utterance U_t (i.e., $R_{t-1}U_t = w_{x_1}, w_{x_2}, ..., w_{x_{2N}}$) with gated recurrent unit (GRU) encoders [8]:

$$\mathbf{h}_1^{(x)}, \mathbf{h}_2^{(x)}, ..., \mathbf{h}_{2N}^{(x)} = GRU(\mathbf{w}_{x_1}, \mathbf{w}_{x_2}, ..., \mathbf{w}_{x_{2N}}), \quad (3)$$

At t-th turn, each token in S_t and R_t is decoded with both probability of direct generation and that of copying from a previous sequence.

We employ an attention decoder [2] to calculate the generation probability of S_t and R_t individually. When decoding an output $Y \in \{S_t, R_t\}$ at t-th turn, the decoder attends back to hidden states $\mathbf{h}^{(x)}$ of the input X to compute attention vectors. We calculate attention vectors with $R_{t-1}U_t$ when generating S_t, and concatenate S_t to $R_{t-1}U_t$ when generating R_t. For the generation of the j-th word in Y, y_j, we calculate its attention score a_{ij} as follows [1]:

$$a_{ij} = softmax(\mathbf{v_1}^T tanh(\mathbf{W_1}\mathbf{h}_i^{(x)} + \mathbf{W_2}\mathbf{h}_{j-1}^{(y)})) \quad (4)$$

where $\mathbf{h}_i^{(x)}$ and $\mathbf{h}_{j-1}^{(y)}$ indicate a hidden state of i-th word in X and a hidden state of $j-1$-th word in Y, respectively. $\mathbf{v_1}$, $\mathbf{W_1}$ and $\mathbf{W_2}$ are learnable parameters. Given a_{ij}, the decoder then generates a hidden representation $\mathbf{h}_j^{(y)}$ of the j-th word using GRU:

$$\tilde{\mathbf{h}}_j^{(x)} = \sum_i a_{ij}\mathbf{h}_i^{(x)},$$

$$h_j^{(y)} = GRU(w_{y_{j-1}}, h_{j-1}^{(y)}, \tilde{\mathbf{h}}^{(x)}), \quad (5)$$

where $\tilde{\mathbf{h}}^{(x)}$ refers to an attention vector by summing up all $\mathbf{h}_i^{(x)}$, and $w_{y_{j-1}}$ indicates the $j-1$ word in Y. Thereafter, we get the generation probability distribution $\mathbf{p}_j^g$ for the j-th output word:

$$\mathbf{p}_j^g = \frac{1}{Z}e^{(\mathbf{W_3}\mathbf{h}_j^{(y)})}, \quad (6)$$

where Z is a shared normalization term and $\mathbf{W_3}$ is a learnable parameter. $\mathbf{p}_j^g$ can be considered as the normalized projection of $\mathbf{h}_j^{(y)}$ at the output space.

---

[1]We also tried dot-product attention and got similar experimental results.

We employ copyNets [14] to compute the probability of copying words from a deterministic input X when decoding y_j. We denote the probability as $p^{c(X)}(y_j)$, and the aggregated probability distribution as $\mathbf{p}_j^{c(X)}$. This probability of generating an word output y_j by copying from X is calculated as:

$$p^{c(X)}(y_j) = \begin{cases} \frac{1}{Z} \sum_{i:w_{x_i}=y_j} e^{\psi(w_{x_i})}, y_j \in X \\ 0, otherwise \end{cases} \qquad (7)$$

where Z is a shared normalization term. $\psi(x_i)$ is the weight of copying the i-th word in X, calculated in Eq. 8:

$$\psi(y_j = w_{x_i}) = \mathbf{v_2}^T tanh(\mathbf{W_4}\mathbf{h}_i^{(x)} + \mathbf{W_5}\mathbf{h}_j^{(y)}), \qquad (8)$$

where $\mathbf{v_2}$, $\mathbf{W_4}$ and $\mathbf{W_5}$ are learnable parameters.

Since S_t could be unlabeled, we may have to copy from a nondeterministic model-generated distribution. Also to enable multiple-step copying along dialogue turns, we propose *implicit copyNets* to copy words with high confidence from a sequence of nondeterministic word distribution. The copying probability is calculated as:

$$p^{c(X)}(y_j) = \frac{1}{Z} \sum_{i=1}^{|X|} p_i(w_{x_i} = y_j)e^{\psi(w_{x_i})}, \qquad (9)$$

where $p_i(w_{x_i} = y_j)$ is the probability that the i-th word in the nondeterministic input X equals y_j. Note that in cases where the source sequence is deterministic such as U_t, $p_i(w_{x_i} = y_j)$ degenerates to a binary indicator where $p_i(w_{x_i}) = 1$ if $w_{x_i} = y_j$. At such situation, implicit CopyNets are equivalent to original copyNets.

Eventually, we demonstrate organization details of CopyFlowNet. A *copy flow* from X to Y refers to a procedure involving copying probability from X at the generation of Y. Specifically, at a single turn t in our model, there are copy flows from $R_{t-1}U_t$ to S_t and from S_t to R_t; there is another copy flow from S_{t-1} to S_t in adjacent dialogue turns. The name of *copyflows* is illuminated by the phenomenon that the information flow of $S_1, S_2, ..., S_t$ relay along the dialogue turns by copying and finally attend to the generation of R_t. Formally, the probability distribution of the j-th word in S_t and R_t are calculated as:

$$\mathbf{p}_j(S_t) = \begin{cases} \mathbf{p}_j^g(S_t) + \mathbf{p}_j^{c(S_{t-1})}(S_t) + \mathbf{p}_j^{c(R_{t-1}U_t)}(S_t), t > 0 \\ \mathbf{p}_j^g(S_t) + \mathbf{p}_j^{c(R_{t-1}U_t)}(S_t), t = 0 \end{cases} \qquad (10)$$
$$\mathbf{p}_j(R_t) = \mathbf{p}_j^g(R_t) + \mathbf{p}_j^{c(S_t)}(R_t).$$

Every copy flow has an intuitive explanation. Copy flows from $R_{t-1}U_t$ to the state span S_t and from S_t to the response R_t enable the model to cache "keywords" in S_t, and then copy them again to the final response R_t. It is inspired from the observation that the keywords indicating dialogue states are prone to co-occur among user inputs and system responses. Given an example dialogue turn:

- User: I want to book a French restaurant.
- Model: Côte Brasserie is a French restaurant with good reputation.

The word "French" is the key word between the utterances which should be copied from U_t to S_t and then be copied to R_t. The model can also generate new words from the full vocabulary with normal

generation process at S_t for further copying to R_t. Note that all these actions can be learned even in unsupervised settings. Besides, the copy flow from the previous state span S_{t-1} to current state span S_t encourages useful dialogue states to be passed through dialogue turns, which intrinsically provides a solution for long-term dependency, as well as enables copying-mechanism to learn information from co-occurrence of keywords across different dialogue turns.

4.3 Posterior regularization

Due to the scarcity of supervised signal in state span S_t, the training of the state span could be unstable, especially when S_t is not annotated. To tackle this problem, we apply posterior regularization on S_t to train the state tracker more stably.

The normal forward pass of the network in the previous discussion parameterized by Θ actually computes the prior probability distribution of S_t:

$$P_\Theta(S_t|R_{t-1}, S_{t-1}, U_t) = \Pi_i \mathbf{p}(s_t^{(i)}|s_t^{(<i)}, R_{t-1}, S_{t-1}, U_t), \qquad (11)$$

We then build another network, the posterior network, which learns the posterior distribution of S_t with more informative inputs. It adopts the same structure but it is separately parameterized [2] with Φ. It takes R_{t-1}, S_{t-1}, U_t, and R_t as input, and calculates the posterior distribution of S_t as follows:

$$Q_\Phi(S_t|S_{t-1}, R_{t-1}, U_t, R_t) = \Pi_i \mathbf{q}(s_t^{(i)}|s_t^{(<i)}, R_{t-1}, S_{t-1}, U_t, R_t), \qquad (12)$$

where R_t is concatenated to the input at the encoder. Note that only the prior network works during testing time, while the the posterior network only directs the prior network in training.

The idea of posterior regularization is to force the prior distribution P to approximate the posterior distribution Q learned from more informative inputs. We utilize KL-divergence to regularize these two distributions. Given multinomial distributions $\mathbf{p}_i$ and $\mathbf{q}_i$ over the vocabulary space, the KL-divergence from $\mathbf{p}_i$ to $\mathbf{q}_i$ is calculated as follows:

$$KL(\mathbf{q}_i||\mathbf{p}_i) = \sum_l^{|\mathcal{V}|} q_i^{(l)} log(\frac{q_i^{(l)}}{p_i^{(l)}}), \qquad (13)$$

The training process of the model varies according to the amount of labeled data. In supervised or semi-supervised learning, we maximize the joint log-likelihood [48] for response and state span generation along with posterior regularization. The learning objective thus comprises three sub-objectives.

$$\mathcal{L}_1 = - \sum^{\mathcal{A}\cup\mathcal{U}} log[P(R_t|R_{t-1}, U_t, S_t)]$$
$$- \sum^{\mathcal{A}} log[P_\Theta(S_t|R_{t-1}, U_t, S_{t-1})Q_\Phi(S_t|R_{t-1}, U_t, S_{t-1}, R_t)] \quad (14)$$
$$+ \lambda \sum^{\mathcal{U}} \sum_{i=1}^{N} KL(\mathbf{q}_i||\mathbf{p}_i),$$

---

[2]The response decoder is shared between two networks in semi-supervised learning senarios.

where $\mathcal{A}$ and $\mathcal{U}$ denote annotated and unannotated training instances respectively. N is the length of the state span.

However, maximum likelihood estimation on the distribution $Q_\Phi(S_t|S_{t-1}, R_{t-1}, U_t, R_t)$ can not be applied when S_t is completely unannotated. We therefore indirectly train the posterior distribution $Q_\Phi(\hat{S}_t|S_{t-1}, R_{t-1}, U_t, R_t)$ with different generation objectives at the decoder of the posterior network. In detail, we feed R_{t-1}, U_t, R_t at its encoder and train the model to reconstruct them at its decoder. The posterior network thereby learns to cache most informative words in both contexts and responses at $\hat{S}_t$ via a structure of auto-encoders. The prior distribution $P_\Theta(S_t|R_{t-1}, S_{t-1}, U_t)$ is regularized towards this posterior distribution with KL-divergence, and the learning objective can be written as

$$
\begin{aligned}
\mathcal{L}_2 = &- \sum^{\mathcal{U}} log[P(R_t|R_{t-1}, U_t, S_t)] \\
&- \sum^{\mathcal{U}} log[Q_\Phi(R_{t-1}, U_t, R_t|\hat{S}_t)] \\
&+ \lambda \sum^{\mathcal{U}} \sum_{i=1}^{N} KL(\mathbf{q}_i||\mathbf{p}_i).
\end{aligned}
\tag{15}
$$

which can be interpreted as response generation loss, reconstruction loss and regularization loss respectively . Following [16], we employ a factor λ as a trade-off factor. See §5.5 for more details.

5 EXPERIMENTAL SETUP

5.1 Research questions

We list the research questions that guide the remainder of the paper: (1) **RQ1**: what is the overall performance of SEDST in task-oriented neural dialogue generation? (See §6.1) (2) **RQ2**: How much does unlabeled data help dialogue state tracking in task-oriented dialogues in our model? (See §6.1.) (2) **RQ3**: Is our explicit state tracker helpful in response generation in non-task-oriented dialogues? (See §6.2) (4) **RQ4**: Does posterior regularization improve the model performance? (See §6.3) (5) **RQ5**: Can our dialogue state tracker generate explainable and representative words? Can it tackle the long-term dependency in a dialogue generation? (See §6.4)

Next, we introduce the datasets in §5.2. The baselines are listed in §5.3 and evaluation methods are depicted in §5.4. Details of the training setting are described in §5.5.

5.2 Datasets

In order to answer our research questions, we work with two task-oriented dialogue corpora: Cambridge Restaurant Corpus and Stanford In-Car Personal Assistant Corpus; two non-task-oriented dialogue corpora: Ubuntu Technical Corpus and JD.com Customer Service Corpus. Details of our datasets are described as follows:

Cambridge Restaurant Corpus Cambridge Restaurant corpus is used to design a dialogue system to assist users to find a restaurant in the Cambridge, UK area [50]. Customers can use three informable slots (*food, pricerange, area*) to constrain the search. This dataset contains 99 restaurants, and 676 clean dialogues out of 1500 dialogue turns. There are 99 possible informable slot values. We split the corpus by 3:1:1 as training, validation and test sets.

Stanford In-Car Personal Assistant Corpus The Stanford driver and car assistant corpus is a multi-turn multi-domain task-oriented dialogue dataset[3]. This dataset includes three distinct domains: calendar scheduling, weather information retrieval, and point-of-interest navigation. There are two modes, namely *Driver* and *Car Assistant*. There are 284 informable slot values for state tracking. Each dialogue is associated with a separate knowledge base (KB) with about 7 entries. The corpus contains 2425 ,302, 302 dialogues for training, validation and testing.

Ubuntu Technical Corpus Ubuntu Dialogue Corpus [27] is an English multi-turn dialogue corpus containing about 487337 dialogues extracted from the Ubuntu Internet Relayed Chat channel. A conversation begins with an Ubuntu-related technical problem, and follows by the responses to the questions. The corpus consists of 448833, 19584, 18920 dialogues of training, validation, testing, respectively. Though the corpus is domain specific, the task and slot-values are not explicitly specified.

JD.com Customer Service Corpus JD.com customer service corpus [7] is a large real-world dataset for online shopping after-sale service. The conversation is between a customer and a customer service staff. It contains 415, 000 dialogues for training, 1, 5000 dialogues for validation, and 5, 005 for testing. We exclude template dialogue turns that the staff or customers merely thanks with keyword filtering.

5.3 Baselines and comparisons

We list the methods and baselines below. We write **SEDST** for the overall process as described in §4, which includes posterior regularization. We write **SEDST\PR** for the model that skips the posterior regularization process.

To assess the contribution of our proposed methods, our baselines include recent work on both task-oriented dialogue models and non-task-oriented dialogue models. We adopt the following baselines for task-oriented dialogue generation under fully supervision.

- **NDM**: Network based Dialogue Models [50] with a CNN-RNN dialogue state tracker
- **NBT**: Neural Belief Tracker [29] with a CNN feature extractor.
- **KVRN**: Key-value retrieval dialogue model [10], which does not adopt a dialogue state tracker but directly retrieves an entry from a key-value structured knowledge base with attention mechanism, and decodes a special token in <subject, relation, object> form during response generation.

We utilize the following representative baselines for non-task-oriented dialogue generation:

- **SEQ2SEQ**: sequence-to-sequence model, also known as recurrent encoder-decoder model [41, 45].
- **HRED**: hierarchical recurrent encoder-decoder model.[43]
- **VHRED**: latent variable hierarchical recurrent encoder-decoder model. [38].
- **HVMN**: hierarchical variational memory network. [7]

---

[3]https://nlp.stanford.edu/blog/a-new-multi-turn-multi-domain-task-oriented-dialogue-dataset/

Table 2: RQ1&RQ2&RQ4: Performance on Cambridge Restaurant corpus. N/A indicates the model fails to produce valid result on this dataset. Best performance is marked bold for each supervision proportion

	Supervision Proportion	BLEU	Joint Goal Accuracy	Entity Match Rate
SEDST	0%	**0.201**	**0.684**	**0.649**
	25%	**0.225**	**0.867**	**0.858**
	50%	**0.236**	**0.945**	**0.927**
SEDST\PR	0%	0.199	0.679	0.422
	25%	0.213	0.854	0.848
	50%	0.192	0.911	0.901
SEDST(without unlabeled data)	0%	-	-	-
	25%	0.091	0.827	0.807
	50%	0.122	0.896	0.899
SEDST(fully supervised)	100%	**0.244**	**0.962**	**0.955**
NDM	100%	0.239	0.921	0.902
Neural Belief Tracker	100%	-	0.865	-
KVRN	100%	0.134	-	N/A

Table 3: RQ1&RQ2&RQ4: Performance on Stanford In-Car Personal Assistant Corpus. Best performance is marked bold for each supervision proportion

	Supervision Proportion	BLEU	Joint Goal Accuracy	Entity Match Rate
SEDST	0%	**0.202**	**0.635**	**0.642**
	25%	**0.192**	**0.758**	**0.813**
	50%	**0.195**	**0.796**	**0.833**
SEDST\PR	0%	0.193	0.622	0.564
	25%	0.180	0.726	0.770
	50%	0.178	**0.796**	0.812
SEDST(without unlabeled data)	0%	-	-	-
	25%	0.102	0.727	0.751
	50%	0.156	0.772	0.773
SEDST(fully supervised)	100%	**0.193**	0.829	**0.845**
NDM	100%	0.186	0.750	0.716
Neural Belief Tracker	100%	-	0.756	-
KVRN	100%	0.172	-	0.459

- **DAWnet**: deep and wide neural network for dialogue generation. It first generates keywords that deepen or widen topics before response generation. Ground truth of keywords are obtained with rules and unsupervised methods [46]

5.4 Evaluation metrics

Task-oriented dialogue evaluation

To assess the language quality and the state tracking ability for the task-oriented dialogue generation, we employ **BLEU** [30], a word-overlapping based metric for language quality evaluation, to measure performance.

To measure the state tracker performance, we employ **Joint Goal Accuracy** [29] as our turn-level evaluation metric. Joint goal accuracy calculates the proportion of the dialogue turns where all the constraints are captured correctly, excluding those where the user merely thanks without extra information.

However, this metric is not applicable for task-oriented dialogue systems without a separate state tracker, such as KVRN. Thus we employ **Entity Match Rate** [50], as an evaluation metric. Entity Match Rate calculates the proportion of the dialogues where all the

constraints are correctly identified when the last placeholder (e.g., poi_SLOT) appears. We skip dialogues without a single placeholder in ground-truth responses; we consider dialogues as failures if no placeholder is decoded for remaining dialogues.

Non-task oriented dialogue evaluation

Evaluating dialogue systems in such a large corpus is not a trivial task. Liu et al. [26] showed that word-overlap automatic metrics like **BLEU** [30] or **ROUGE** [25] are not well correlated with human evaluations regarding response quality. To evaluate the semantic relevance between the candidate response and target response, we employ three embedding-based topic similarity metrics proposed by Liu et al. [26]: **Embedding Average**, **Embedding Extrema** and **Embedding Greedy** [11, 28, 37]. We employ publicly available *word2vec*[4] to train word embeddings for evaluation. We train English word embeddings on Google News Corpus. For Chinese, the word embeddings are trained on Chinese Giga-word corpus version 5 [13], segmented by *zpar*[5] [60].

---

[4]https://code.google.com/archive/p/word2vec/
[5]https://github.com/SUTDNLP/ZPar

5.5 Experimental settings

On Cambridge Restaurant corpus and Stanford In-Car Personal Assistant corpus, we trained the model with Adam [17] optimizer with a learning rate of 0.003 with early stopping. The batch-size was set to 32, and the size of the word embedding was set to 50. We used a single-layer GRU with 50 hidden units. The trade-off factor λ was set to 0.1. For non-task-oriented dialogue models on Ubuntu Dialogue Corpus and JD.com Customer Service Corpus, we trained the model with Adam [17] optimizer, under a learning rate of 0.0005 with early stopping. The batch size was set to 24 and the size of the word embedding was set to 300. We used a single-layer GRU with 500 hidden units for these models. The trade-off factor λ was set to 0.1 at the beginning, and uniformly decrease to 0.001 within the first epoch. We loaded pretrained fastText [4] word vectors for all the models, and the vocabulary size on these datasets was limited to 800, 1400, 20000, and 20000, respectively.

For response generation, we applied beam search decoding with a beam size of 5. As for dialogue state decoding in semi-supervised or fully-supervised scenario, we performed max-sampling on $\mathbf{p}_i$ to generate word sequences in state tracker with a special token indicating sequence termination. However, in unsupervised scenario, it is impossible for model to generate end-of-sequence tokens. Thus, our state decoder generated a sequence by a fixed time step of T_s during decoding in unsupervised setting. In our experiments, T_s was set to 5 on Ubuntu Technical Dialogue Corpus and 8 for all other corpora. We prevent generation of repeated words in the state span during sampling. During the evaluation of unsupervised dialogue state trackers on task-oriented corpora, we calculated the intersection of the output of state decoders and all possible slot values provided separately in the corpus. This setup is same as baselines where dialogue states are limited to a fix-sized set.

6 EXPERIMENTAL RESULTS

In §6.1, we compare our methods to baselines for task-oriented dialogue generation; in §6.2 we examine the performance of comparisons for non-task-oriented dialogue generation; §6.3 examines the effect of posterior regularization. We discuss the explainability and representativeness of dialogue state tracking in §6.4.

6.1 Task-oriented dialogue systems

To start, we address research question **RQ1** for task-oriented dialogue systems. Table 2 and Table 3 list the performance of all methods on two task-oriented dialogue corpora respectively. For all two datasets, under fully supervised settings, SEDST outperforms other baselines and achieves state-of-the-art performance. On Cambridge corpus, SEDST with fully supervision achieves a 2.09%, 4.45%, and 5.87% over NDM in terms of BLEU, joint goal accuracy, and entity math rate, respectively; whereas on Stanford corpus, it achieves a 3.76%, 10.5%, and 18.0%, respectively.

To address research question **RQ2**, we analyze the performance of our models in semi-supervised settings where only part of labeled data is available. We find that SEDST outperforms SEDST trained without unlabeled data in both corpora on all corpora. In details, when only 25% of data is labeled, in terms of joint goal accuracy, SEDST offers a 4.0% and 5.1% increase on two corpora respectively; whereas it gives 3.1% and 6.2% increase in terms of entity match

rate. When only 50% of data is labeled, its increase become 4.9%, 2.8% and 2.4%, 6.0% respectively. It verifies that our model is capable to utilize unlabeled data well for training dialogue state trackers. Moreover, we find our model outperforms state-of-the-art baselines on two corpora for both state tracking ability and language quality when only 50% of labeled data is available. We notice that SEDST provides fairly satisfying results even if the state tracker is trained in a complete unsupervised manner. In terms of joint goal accuracy, SEDST achieves 68.4% and 63.5% on two corpora respectively; whereas entity match rate performance becomes 64.9% and 64.2%.

6.2 Non-task oriented dialogue systems

Next, we turn to **RQ3**. From table 4 and table 5, We find *SEDST* produces quite competitive results against state-of-the-art non-task-oriented dialogue models. Quite notably, on all corpora SEDST outperforms VHRED and HVMN, which uses continuous latent variables to maintain dialogue states. In addition, our state tracker produces explicit dialogue states in state trackers. We notice that the entities mentioned between context and generated responses are highly relevant, which effectively tackles a key challenge in neural dialogue generation. Table 8 provides examples of state tracker outputs and generated responses.

6.3 Effect of posterior regularization

Turning to **RQ4**, shown in Table 2 and Table 3, we find that our posterior regularization improves overall state tracking performance from the comparison between SEDST and SEDST\PR. On Cambridge corpus, when only 25% labeled data is available, SEDST offers a 1.3% and 1.0% increase over SEDST\PR in terms of joint goal accuracy and entity match rate respectively; while it gives a 3.4%, 2.6% increase when 50% labeled data is available. For Stanford corpus, we see a similar picture. It also improves response quality, as shown in Table 4 and Table 5.

Here we discuss the effect of posterior regularization. In terms of state tracking in task-oriented dialogues, we train the posterior network with more informative input by including current turn response R_t; then we optimize the prior network by minimizing the distance of posterior and prior distributions (Eq. 13). Accordingly, SEDST performs better than SEDST\PR when fewer labeled data is available. Posterior regularization also helps the response generation with unlabeled dialogues states. Although the prior network can explore a generation strategy of S_t, S_t is also regularized towards the compressed representation of context and responses learned by the posterior network. The representation which we regularize towards is a probability distribution over the vocabulary space learned by auto-encoders. We notice a state-of-the-art work, DAWnet [46], extracts keywords with rules and unsupervised methods from unlabeled data. Key word prediction is then trained with supervised methods with maximum likely hood estimation objective in the neural network. However, posterior regularization is a generalization of the learning strategy of DAWnet [46]. When the i-th word over the vocabulary is treated as the ground truth of the keyword, the objective of maximum likelihood estimation is equivalent to minimizing the KL-divergence between the prior distribution $\mathbf{p}$ and an one-hot distribution $\hat{\mathbf{q}}$, since we have:

$$log(p_i) = -KL(\hat{\mathbf{q}}||\mathbf{p}). \tag{16}$$

Table 4: RQ3: Embedding-based evaluation in Ubuntu Technical Corpus. Emb. is an abbreviation for Embedding

Model	Emb. Average	Emb. Greedy	Emb. Extrema
SEQ2SEQ	0.216	0.169	0.126
HRED	0.542	0.412	0.319
VHRED	0.534	0.403	0.306
HVMN	0.558	0.423	0.322
DAWnet	0.530	0.390	0.333
SEDST\PR	0.586	0.438	0.330
SEDST	**0.609**	**0.451**	**0.337**

Table 5: RQ3: Embedding-based evaluation in JD Corpus. Emb. is an abbreviation for Embedding

Model	Emb. Average	Emb. Greedy	Emb. Extrema
SEQ2SEQ	0.425	0.479	0.264
HRED	0.549	0.587	**0.406**
VHRED	0.576	0.593	0.392
HVMN	0.564	0.596	0.405
DAWnet	0.579	0.574	0.375
SEDST\PR	0.575	0.602	0.373
SEDST	**0.585**	**0.607**	0.392

Table 6: RQ4: Proportion of generated *predicted keywords* [46] existing in ground truth responses.

Model	Ubuntu Technical	Jd.com Customer Service
DAWnet	5.5%	32.6%
SEDST	14.7%	40.6%

where $\hat{q}$ is a multinomial distribution with $\hat{q}_i = 1$ at its i-th coordinate. However, unlike DAWnet, posterior regularization provides probability distributions over the whole vocabulary space as learning signals, which are more informative than a handful of keywords and immune to extraction bias from rules. Moreover, our model is fully end-to-end trainable.

We notice that the text spans in DAWnet are trained on *predicted keywords* [46], which are defined as the keywords that appear in ground truth responses but not appear in contexts. As an empirical study of the contribution of S_t, in Table 6 we present the proportion of generated *predicted keywords* that exist in ground truth responses. In contrast with DAWnet, SEDST gives an obviously larger proportion of correct *predicted keywords* for both corpora.

6.4 Case study

Finally, we address **RQ5**. In Table 7 and 8 we provide example dialogues in task-oriented dialogue corpus and non-task-oriented dialogue corpus under unsupervised training. We see our model successfully captures informative words indicating dialogue states without any supervision. Especially in examples from the Ubuntu corpus, our state tracker generates explainable and representative words of dialogue process rather than generic words like "the" or "is", which co-occurs much more often in the whole corpus. Some important words are passed through dialogue turns, indicating that explicit tracking of dialogue states with copying mechanism is effective for tackling long-term dependency. The representative words are also addressed as *topic continuity* [21] between turns.

7 CONCLUSION

We have addressed the task of explicit dialogue state tracking for both task-oriented and non-task-oriented dialogue generation. We have identified challenges of dialogue state tracking: expensive nature of state labeling and weak explainability. To tackle these challenges, we have proposed a semi-supervised neural dialogue generation framework, named SEDST. Along with SEDST, we have proposed an explicit dialogue state tracking model, CopyFlowNet, including an implicit copyNet mechanism for representing explicit word sequences and a posterior regularization strategy for semi-supervised learning procedure. In our experiments, we have presented a detailed comparison and deep analysis of the performance of SEDST, and have verified its effectiveness in terms of the dialogue generation quality and the dialogue state tracking accuracy on both non-tasked oriented dialogue corpora and task-oriented dialogue corpora.

As to future work, we plan to transfer our model to other tasks such as reading comprehension, filtering, and summarization [22, 33, 35]. Also, we would like to apply reinforcement learning to improve the performance of dialogue generation.

REFERENCES

[1] D. Ameixa, L. Coheur, P. Fialho, and P. Quaresma. *Luke, I am Your Father: Dealing with Out-of-Domain Requests by Using Movies Subtitles.* Springer International Publishing, 2014.
[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.
[3] R. E. Banchs and H. Li. Iris: a chat-oriented dialogue system based on the vector space model. In *ACL*, 2013.
[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *TACL*, 5, 2017.
[5] A. Bordes, Y. L. Boureau, and J. Weston. Learning end-to-end goal-oriented dialog. In *ICLR*, 2017.
[6] K. Cao and S. Clark. Latent variable dialogue models and their diversity. In *EACL: Volume 2, Short Papers*, 2017.
[7] H. Chen, Z. Ren, J. Tang, E. Y. Zhao, and D. Yin. Hierarchical variational memory network for dialogue generation. In *WWW*, pages 1653–1662, 2018.
[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *arXiv preprint arXiv:1412.3555*, 2014.
[9] B. Dhingra, L. Li, X. Li, J. Gao, Y.-N. Chen, F. Ahmed, and L. Deng. Towards end-to-end reinforcement learning of dialogue agents for information access. In *ACL*, 2017.
[10] M. Eric and C. D. Manning. Key-value retrieval networks for task-oriented dialogue. In *SIGDIAL*, 2017.
[11] G. Forgues, J. Pineau, J.-M. Larchevêque, and R. Tremblay. Bootstrapping dialog systems with word embeddings. In *NIPS, Modern Machine Learning and Natural Language Processing Workshop*, 2014.
[12] D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai. A form-based dialogue manager for spoken language applications. In *ICSLP*, 1996.
[13] D. Graff and K. Chen. Chinese gigaword. *LDC Catalog No.: LDC2003T09, ISBN*, 1, 2005.
[14] J. Gu, Z. Lu, H. Li, and V. O. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*, 2016.
[15] M. Henderson, B. Thomson, and S. Young. Deep neural network approach for the dialog state tracking challenge. In *SIGDIAL*, 2013.
[16] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.
[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
[18] S. Lee. Structured discriminative model for dialog state tracking. In *SIGDIAL*, 2013.
[19] S. Lee and M. Eskenazi. Recipe for building robust spoken dialog state trackers: Dialog state tracking challenge system description. In *SIGDIAL*, 2013.
[20] W. Lei, X. Jin, Z. Ren, M.-Y. Kan, X. He, and D. Yin. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *ACL*, 2018.
[21] W. Lei, Y. Xiang, Y. Wang, M. Liu, and M.-Y. Kan. Linguistic properties matter for implicit discourse relation recognition: Combining semantic interaction, topic

Table 7: Examples task-oriented dialogues in unsupervised training on Standford In-Car Assistant Corpus and Cambridge Restaurant Corpus

Contexts	Generated	Ground Truth
i am looking for a german restaurant in the west part of town .	**State**: german west **Generated**: there are no german restaurants in the west part of town . would you like to try something else ?	**State**: west german **Response**: there are no german restaurants in the west part of town . would you like to make another query ?
how about italian ?	**State**: italian west **Generated**: name_SLOT is located in the west part of town .	**State**: italian west **Response**: there are several italian restaurants in the west part of town . name_SLOT is pricerange_SLOT and name_SLOT ispricerange_SLOT . would you like the address of one of those ?
what is the phone number of one of them ?	**State**: italian west **Generated**:name_SLOT is located at address_SLOT and their phone number is phone_SLOT .	**State**: italian west **Response**: the phone number to name_SLOT , which is pricerange_SLOT priced , is phone_SLOT . is there anything else i can assist you with today ?

Table 8: Examples dialogues on Ubuntu Technical corpus. The words that co-occur between generated dialogue state spans and responses are marked in italic

Contexts	Generated	Ground Truth
I just realised nothing can access my mysql database anymore , it says cannot connect , why would this be ? I did set it to be remote	**State**: remote would anymore mysql access **Response**: you can use the *mysql* client to connect to the server	**State**: - **Response**: is it running ?
is it running ?	**State**: running I says mysql access **Response**: yes , *I* am *running* it now	**State**: - **Response**: How do *I* check ? I was able to type *mysql* commands
How do I check ? I was able to type mysql commands	**State**: commands mysql I running How **Response**: you can use sudo	**State**: - **Response**: thats not mean your server was *running* check with ps or top or others tool
thats not mean your server was running check with ps or top or others tool	**State**: mysql top tool server check **Response**: thanks	**State**: - **Response**: How do I *check* if it 's running ? Ah , k
How do I check if it 's running ? Ah , k	**State**:mysql check ps server thats **Response**: *ps* aux | grep *mysql*	**State**: - **Response**:*ps* aux | grep mysqld

continuity and attribution. In *AAAI*, 2018.

[22] C. Li, W. Zhou, F. Ji, Y. Duan, and H. Chen. A deep relevance model for zero-shot document filtering. In *ACL*, pages 2300–2310, 2018.

[23] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. In *NAACL*, 2016.

[24] J. Li, M. Galley, C. Brockett, G. Spithourakis, J. Gao, and B. Dolan. A persona-based neural conversation model. In *ACL*, 2016.

[25] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 2004.

[26] C. W. Liu, R. Lowe, I. Serban, M. Noseworthy, L. Charlin, and J. Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*, 2016.

[27] R. Lowe, N. Pow, I. Serban, and J. Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *SIGDIAL*, 2015.

[28] J. Mitchell and M. Lapata. Vector-based models of semantic composition. In *ACL*, 2008.

[29] N. Mrkšić, D. O. Séaghdha, T.-H. Wen, B. Thomson, and S. Young. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*, 2016.

[30] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 50th Annual Meeting on Association for Computational Linguistics*, 2002.

[31] J. Pei and C. Li. S2spmn:a simple and effective framework for response generation with relevant information. In *EMNLP*, 2018.

[32] J. Perez and F. Liu. Dialog state tracking, a machine reading approach using memory network. In *EACL*, 2017.

[33] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[34] H. Ren, W. Xu, Y. Zhang, and Y. Yan. Dialog state tracking using conditional random fields. In *SIGDIAL*, 2013.

[35] Z. Ren, O. Inel, L. Aroyo, and M. De Rijke. Time-aware multi-viewpoint summarization of multilingual social text streams. In *CIKM*, pages 387–396, 2016.

[36] A. Ritter, C. Cherry, and W. B. Dolan. Data-driven response generation in social media. In *EMNLP*, 2011.

[37] V. Rus and M. Lintean. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, 2012.

[38] I. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, 2017.

[39] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, 2016.

[40] I. V. Serban, T. Klinger, G. Tesauro, K. Talamadupula, B. Zhou, Y. Bengio, and A. C. Courville. Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*, 2017.

[41] L. Shang, Z. Lu, and H. Li. Neural responding machine for short-text conversation. In *ACL, IJCNLL*, Beijing, China, 2015.

[42] B. A. Shawar and E. Atwell. Chatbots: are they really useful? *Ldv Forum*, 22(1), 2007.

[43] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, and J.-Y. Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *CIKM*, 2015.

[44] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan. A neural network approach to context-sensitive generation of conversational responses. In *NAACL-HLT*, Denver, Colorado, 2015.

[45] O. Vinyals and Q. Le. A neural conversational model. In *The 32nd International Conference on Machine Learning Deep Learning Workshop*, 2015.

[46] W. Wang, M. Huang, X.-S. Xu, F. Shen, and L. Nie. Chat more: Deepening and widening the chatting topic via a deep model. In *SIGIR*, 2018.

[47] Z. Wang and O. Lemon. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *SIGDIAL*, 2013.

[48] T.-H. Wen, Y. Miao, P. Blunsom, and S. Young. Latent intention dialogue models. *arXiv preprint arXiv:1705.10229*, 2017.

[49] T.-H. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*, 2017.

[50] T.-H. Wen, D. Vandyke, N. Mrkšić, M. Gasic, L. M. Rojas Barahona, P.-H. Su, S. Ultes, and S. Young. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*, 2017.

[51] J. Williams. Multi-domain learning and generalization in dialog state tracking. In *SIGDIAL*, 2013.

[52] J. Williams, A. Raux, D. Ramachandran, and A. Black. The dialog state tracking challenge. In *SIGDIAL*, 2013.

[53] J. D. Williams. A belief tracking challenge task for spoken dialog systems. In *NAACL-HLT Workshop on Future Directions and Needs in the Spoken Dialog Community: Tools and Data*, 2012.

[54] J. D. Williams. Web-style ranking and SLU combination for dialog state tracking. In *SIGDIAL*, 2014.

[55] J. D. Williams, K. Asadi, and G. Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL*, 2017.

[56] C. Xing, W. Wu, Y. Wu, J. Liu, Y. Huang, M. Zhou, and W.-Y. Ma. Topic augmented neural response generation with a joint attention mechanism. *URL http://arxiv.org/abs/1606.08340*, 2016.

[57] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2), 2010.

[58] S. Young, M. Gašić, B. Thomson, and J. D. Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5), 2013.

[59] J. Zhang, Y. Liu, H. Luan, J. Xu, and M. Sun. Prior knowledge integration for neural machine translation using posterior regularization. In *ACL*, 2017.

[60] Y. Zhang and S. Clark. Syntactic processing using the generalized perceptron and beam search. *Computational linguistics*, 37(1), 2011.