Recent Neural Methods on Dialogue State Tracking for Task-Oriented Dialogue Systems: A Survey

Vevake Balaraman[1,2], **Seyedmostafa Sheikhalishahi**[1,2], **Bernardo Magnini**[1]
[1] Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento — Italy
[2] ICT Doctoral School, University of Trento — Italy.
{balaraman,ssheikhalishahi,magnini}@fbk.eu

Abstract

This paper aims at providing a comprehensive overview of recent developments in *dialogue state tracking (DST)* for task-oriented conversational systems. We introduce the task, the main datasets that have been exploited as well as their evaluation metrics, and we analyze several proposed approaches. We distinguish between *static ontology* DST models, which predict a fixed set of dialogue states, and *dynamic ontology* models, which can predict dialogue states even when the ontology changes. We also discuss the model's ability to track either single or multiple domains and to scale to new domains, both in terms of knowledge transfer and zero-shot learning. We cover a period from 2013 to 2020, showing a significant increase of multiple domain methods, most of them utilizing pre-trained language models.

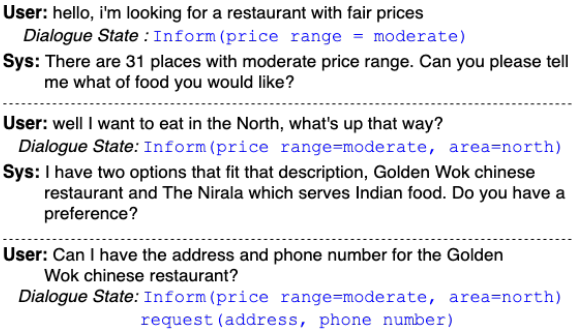
1 Introduction

Task-oriented dialogue systems enable users to accomplish tasks, such as ticket booking, restaurant reservation, and customer support, by interacting in natural language. The ability to accurately track the user's requirements during the dialogue is crucial to enable a consistent and effective dialogue (Wu et al., 2019). Dialogue systems track such information using a dialogue state tracker (DST) component, where a *dialogue state* is represented with slot-value pairs, each denoting a specific user's requirement. The accurate tracking of this information is crucial, as downstream components, like the dialog manager, rely on the dialogue state to choose the next action of the system.

In recent years the performance of several natural language processing (NLP) tasks, including dialogue state tracking (Goldberg, 2017; Chen et al., 2017), has been pushed forward by neural network-based approaches. The DST task actually merges some aspects of natural language understanding in dialogues, although it is more complex than the standard *slot filling* task. In fact, while *slot filling* involves predicting the slot-value pairs referred in a particular turn in dialogue (Louvan and Magnini, 2020), *DST* involves predicting the slot-value pairs at the dialogue level until the current turn. The complexity of DST has driven research to propose various neural approaches, including recurrent neural networks-based (Henderson et al., 2014c; Henderson et al., 2014; Wen et al., 2017; Xu and Hu, 2018; Ren et al., 2018), attention-based models (Wu et al., 2019; Xu and Hu, 2018; Nouri and Hosseini-Asl, 2018), and the very recent transformer-based models (Heck et al., 2020; Kim et al., 2020; Zhang et al., 2019; Lee et al., 2019; Rastogi et al., 2020; Balaraman and Magnini, 2021; Lin et al., 2020). In addition, the rapid progress of NLP has provided technologies to address several DST challenges, including predicting slot-values that are not present in training data, moving from rule-based to learning methods for dialogue state updating, and addressing long-term dependency, a crucial aspect in dialogue. Furthermore, encouraged by the considerable success in modeling single domain dialogues (Henderson et al., 2014c; Wen et al., 2017; Mrkšić et al., 2017a), research on DST has recently moved toward building models that can handle multiple domains (Wu et al., 2019; Zhang et al., 2019; Zhong et al., 2018; Heck et al., 2020), and that are flexible enough to be adapted to new domains (Rastogi et al., 2020; Balaraman and Magnini, 2021; Lin et al., 2020; Gao et al., 2019).

Although such rapid signs of progress have generated an impressive amount of research in DST, including several datasets and experimental material, to the best of our knowledge, such a massive amount of recent work has been only poorly documented (Williams et al., 2016a; Chen et al., 2017), and there is not an updated survey of the field. This paper intends to fill such a gap, providing

Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 239–251
July 29–31, 2021. ©2021 Association for Computational Linguistics



User: hello, i'm looking for a restaurant with fair prices
Dialogue State : `Inform(price range = moderate)`
Sys: There are 31 places with moderate price range. Can you please tell me what of food you would like?

User: well I want to eat in the North, what's up that way?
Dialogue State: `Inform(price range=moderate, area=north)`
Sys: I have two options that fit that description, Golden Wok chinese restaurant and The Nirala which serves Indian food. Do you have a preference?

User: Can I have the address and phone number for the Golden Wok chinese restaurant?
Dialogue State: `Inform(price range=moderate, area=north)`
`request(address, phone number)`

Figure 1: A sample dialogue, from the WoZ2.0 dataset, showing the dialogue states at each user turn.

a comprehensive overview of recent developments in dialogue state tracking applied to task-oriented dialogue systems.

2 Dialogue State Tracking

We first introduce the notion of *dialogue state*, and then describe the *DST* task, giving details on different dialogue state prediction strategies.

Dialogue State. A dialogue state s_t at any turn t in a dialogue comprises the summary of the dialogue history until turn t, such that s_t contains all sufficient information for the system to choose the next action (Williams et al., 2016b). Specifically, it captures the user goals in the conversation in the form of $(slot, value)$ pairs. The set of possible slots is predefined in the Ontology O, typically domain-dependent, while the values assumed by each slot s are provided by the user as a dialogue goal. For example, a dialogue state at turn t in a dialogue for the RESTAURANT domain could be $s_t = \{(\text{FOOD}, \text{ITALIAN}), (\text{AREA}, \text{CENTRE})\}$. This dialogue state encodes the user's goal for slots FOOD and AREA, based on the dialogue history. A slot s can either be of type *informable* or *requestable*. Informable slots are attributes that can be provided by the user during the dialogue as constraints, while requestable slots are attributes that the user may request from the system. In case of the restaurant domain, the slots FOOD, AREA and PRICE are informable, while the slots PHONE and ADDRESS are requestable. Figure 1 shows the tracking of dialogue states at each user turn for the restaurant domain.

Dialogue State Tracker. A DST is responsible for estimating the current dialogue state by predicting the slot-value pairs at turn t. This prediction can be performed in two ways: i) *turn-level prediction*, predicting the slot-values expressed at each turn and then using an update mechanism to combine the previous dialogue state and the current turn prediction; or ii) *dialogue-level prediction*, predicting the complete dialogue state at each turn.

Turn-level prediction. In turn-level prediction the update mechanism can be either rule-based or learned using an update function. In the *rule-based* approach the model makes predictions only for the *slot-values* expressed in the current turn. The dialogue state s_{t-1} from the previous turn $t-1$ and the current turn predictions are then combined using rules to get the current dialogue state s_t. Such rules could either be simple, as combining s_{t-1} and the current turn prediction, with the current turn prediction having the priority (i.e., overwriting values in s_{t-1} if the same slot is expressed in the current turn predictions), or more complex, as using probabilities of the predictions combined with rules to get s_t. In the *learning to update* approach, a function is learned to approximate the update mechanism. It takes the previous dialogue state and the current turn-level prediction as input, and learns how to predict the current dialogue state. This approach can be modelled either with two components or with a single end-to-end model.

Dialogue level prediction. Here, at each turn t of the dialogue, the model takes as input the complete dialogue history and makes predictions for the complete dialogue state s_t. Since the prediction at each turn does not consider the previous dialogue states, this approach has the drawback that the dialogue state at current turns s_t may not be consistent with the preceding dialogue state s_{t-1}.

3 DST Datasets and Evaluation Metrics

In this section we introduce the datasets that have been used in DST in a period from 2013 to 2020, as well as the evaluation metrics for the task.

3.1 Dialog State Tracking Challenge (DSTC)

The dialog state tracking challenge (DSTC) is a series of dialogue related challenges that serves as a common test and evaluation suite for dialogue state tracking (Williams and Young, 2007; Williams et al., 2013, 2016b). The challenge was later renamed as *dialog system technology challenge* to accommodate various other dialogue related tasks. The most widely used datasets in the context of the DST challenge are DSTC2 and DSTC3.

DSTC2 and DSTC3. The dialog state tracking challenges 2 (DSTC2 - (Henderson et al., 2014a)) and 3 (DSTC3 - (Henderson et al., 2014b)) are human-machine conversation dialogue datasets collected using Amazon Mechanical Turk, respectively for the restaurant and the tourist domain.

DSTC2 is a spoken dialogue dataset consisting of automatic speech recognition (ASR) hypotheses and turn-level semantic labels along with the transcriptions. The dataset consists of 1,612 dialogues for training, 506 dialogues for development, and 1,117 dialogues for testing. DSTC3 aims to evaluate DST models on their ability to track unseen slot values and on their adaptability to a new domain. For this purpose, the dataset does not contain training dialogues and consists of 2,265 dialogues for testing. Typically, the models trained on the DSTC2 dataset were evaluated with the DSTC3 dataset to estimate their performance.

3.2 WoZ2.0

The WoZ2.0 dataset was initially published as CamRest dataset with 676 dialogues (Wen et al., 2017). Subsequently, (Mrkšić et al., 2017a) updated CamRest and named it WoZ2.0. The dataset was collected using a Wizard of Oz framework and contains 1,200 dialogues, out of which 600 are for the training set, 200 for the development set, and 400 for the testing set. WoZ2.0 consists of written text conversations for the restaurant booking task. Each turn in a dialogue was contributed by different users, who had to review all previous turns in that dialogue before contributing to the turn. Besides, WoZ2.0 has been translated to Italian and German by professional translators (Mrkšić et al., 2017b).

3.3 MultiWoZ

MultiWoZ is the first widely used multi-domain dialogue dataset for the DST task. It is collected using Wizard-of-Oz and consists of dialogues in 7 domains: restaurant, hotel, attraction, taxi, hospital, and police. 10,438 dialogues were released, out of which 3,406 are single-domain dialogues and 7,032 are multi-domain dialogues (Ramadan et al., 2018). Each of the multi-domain dialogues consists of at least 2 up to 5 domains. MultiWoZ has seen various versions, with several error corrections (Ramadan et al., 2018; Budzianowski et al., 2018; Eric et al., 2020; Zang et al., 2020).

3.4 Schema-Guided Dataset

The schema-guided dataset (SGD) was collected using a bootstrapping approach (Shah et al., 2018), where a dialogue simulator interacts with a service configuration defined by the developer to generate dialogue outlines. The obtained dialogue outlines are then paraphrased using crowd workers. The SGD dataset consists of dialogues in 16 domains for training, 16 domains for development, and 18 domains for testing (Rastogi et al., 2020). Since a domain can be represented by multiple services, the dataset amounts to 26 services in training, 17 services in development, and 21 services in testing. SGD includes 16,142 dialogues for training, 2,482 for development, and 4,201 for testing. The SGD defining feature is the inclusion of new services both in the development (8) and testing (15) sets (all following the same schema structure), which are not present in the training set.

3.5 TreeDST

TreeDST is collected using a bootstrapping approach, with conversations covering 10 domains. A dialogue simulator is used to produce a meaningful conversational flow with a template-based utterance, which is then paraphrased by crowd workers. The dialogue states and the system acts are annotated as tree-structures with hierarchical meaning representations to incorporate semantic compositionality, cross-domain knowledge sharing, and co-reference. The dataset consists of a total of 27,280 conversations (Cheng et al., 2020), which exhibit nested properties for the slots PEOPLE, TIME and LOCATION that are shared across all domains. The dataset also models certain failure situations in the dialogue system, such as glitches (system failures), and uncooperative user behavior.

3.6 Machine-to-Machine

Machine-to-Machine (M2M) dialogues are collected using a bootstrapping approach (Shah et al., 2018) based on dialogue simulators, and are then converted into natural language by crowd workers. The dataset consists of single domain dialogues for restaurant reservation and movie booking including, respectively, 2,240, 768, and 120K dialogues (Shah et al., 2018; Liu et al., 2018).

Among the datasets discussed in this study, DSTC2 and WoZ2.0 are the most used datasets for training single domain models, while MultiWoz is widely used for multi-domain models.

3.7 Evaluation Metrics

The evaluation of dialogue state trackers is performed using automated metrics, namely average goal accuracy, joint goal accuracy, requested slots F1 and time complexity. In the following, a brief description of each metric is provided.

Average Goal Accuracy is the average accuracy of predicting the correct value for a slot, computed only on the informable slots.

Joint Goal Accuracy is the primary evaluation metric for DST. The joint goal is the set of accumulated turn level goals up to a given turn in the dialogue. It indicates the model performance in predicting all slots in a given turn correctly. It is denoted by the fraction of turns in a dialogue where all slots in a turn are predicted correctly.

Requested Slots F1 indicates the model performance in correctly predicting if a requestable slot is requested by the user, estimated as the macro-averaged F1 score over for all requested slots.

Time Complexity denotes the time latency of the model in making predictions. While this metric is not reported for many published studies, given that a dialogue system should respond in real-time, this metric indicates the usability of the model in real-world applications.

4 Static Ontology DST Models

The main distinguishing characteristic of DST models, in our opinion, is their capacity to predict dialogue states either from a fixed set of slot-values (i.e., from a static ontology) or from a possible open set of slot-values (i.e., from a dynamic ontology).

Static ontology models rely on a fixed ontology to predict the dialogue state. This means that the set of slot-values is predefined, and that a model can only predict for those predefined values. These models typically consist of an input layer that transforms each input token into an embedding, of an encoder layer that encodes the input to a hidden state h_t, and of an output layer that predicts the slot value based on h_t. Considering that the set of possible slot-values is predefined, there are two approaches used for the output layer: i) a feed-forward layer, which receives the input representation and produces scores equal to the # of slot-values; ii) an output layer that receives both the input and the slot-value representations and compares them with each of the slot-value representations providing a score for each slot-value. The obtained score can then be normalized using a non-linear activation function, either *softmax*, to get a probability distribution over all the slot-value pairs, or *sigmoid*, to get the individual probability for each slot-value pair. Figure 2 shows the standard architecture of the two approaches.

We now review few challenges that have been addressed in static ontology models, including delexicalization, data-driven DST, parameter sharing, latency in prediction, and the use of pre-trained language models. Performances of the systems are all reported in Table 2.

Delexicalization. *Delexicalization* is an effective approach adopted to counter imbalanced training data for slot-values. In this regard, the slot values in the input are replaced with labels corresponding to slot names. For instance, *I want Chinese food* is delexicalised as *I want F.VALUE F.SLOT*. It has to be noted that replacing slot-values needs a semantic dictionary listing the possible values for each slot. (Henderson et al., 2014c; Henderson et al., 2014) has proposed a word-based DST with recurrent neural networks that uses delexicalization on top of an input representation based on Automatic Speech Recognition. This allows to improve the system robustness with respect to the user expressions mentioning slot values.

Data-driven DST. Although delexicalization showed to be effective, it requires additional manual feature engineering. An alternative, data-driven methodology, was proposed by the neural belief tracker (NBT) (Mrkšić et al., 2017a). Instead of delexicalizing the input, a separate module was learned to represent the slot-value pairs. Then, the slot-value representation and the input representation are passed through a *binary decision maker* before applying *softmax* activation. Similarly, a fully statistical NBT was proposed by (Mrkšić and Vulić, 2018), where a statistical update function replaces the rule-based update mechanism in NBT. The experimental results showed the statistical update function to outperform the rule-based update.

Parameter sharing. While the previous models consist of a separate encoder for each slot whose values have to be predicted, the DST efficiency crucially depends on the number of model parameters. In this direction, (Ren et al., 2018) proposed

Metric	Datasets							
	DSTC2	DSTC3	WoZ2.0	MultiWoZ	Frames	SGD	M2M	TreeDST
# Dialogues	3235	2236	1200	10438	1369	22825	120000	27280
# Turns	51002	35723	8824	143048	19986	463282	1661536	167507*
Avg. turns / dial.	15.77	15.98	7.35	13.7	14.60	20.30	13.85	6.14*
Avg. tokens / turn	8.47	10.82	11.27	13.18	12.60	9.86	9.96	7.59*
# Unique tokens	1178	1873	3562	30245	13864	45578	2315	7936*
# Slots	8	13	7	29	60	339	5	289
# Values	85	118	88	2180	4508	25123	92	5687

*TreeDST provides natural language only for user turns, and not for system acts. No. of turns is computed only on user turns.

Table 1: Statistics of available data sets for the dialogue state tracking task.

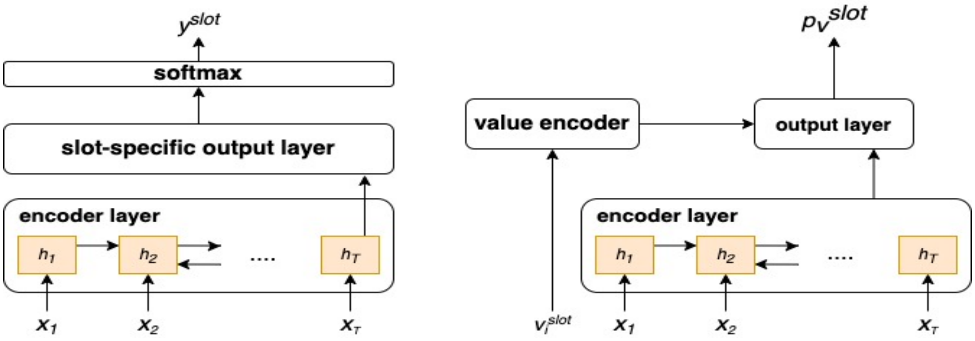


Figure 2: *Left:* model with softmax activation to predict over all slot-values, *Right:* model using value representations to predict the score.

StateNet, a DST sharing the parameters for all slots, thus reducing the number of model parameters. *StateNet* combines a n-gram input feature representation with a slot representation, and uses long short term memory (LSTM) to encode them into a single vector. The value representation is then compared with the encoded vector to obtain the score for each slot-value. A semantically specialised Paragram-SL999 (Wieting et al., 2015) was used to encode the tokens. Compared with fully statistical NBT, *StateNet* achieves high performance even with a rule-based update function.

RNN and latency in DST. A relevant issue for DST models is prediction time, due to the number of dialogue states they have to consider at each dialogue turn. (Zhong et al., 2018) combined both a shared representation and a slot-specific representation in the Global-Locally Self Attentive Dialogue State Tracker (GLAD). The GLAD model consists of an RNN-based global module, to learn global features, and a local module that learns slot-specific features. The representations of slot-values and user input are then scored using a scoring module that predicts their probability. However, GLAD needs an RNN for each slot-value representation, this way increasing the latency of the model. Further improvements on latency were proposed in GCE, Globally-Conditioned Encoder (Nouri and Hosseini-Asl, 2018), which uses only the global encoder, and in (Balaraman and Magnini, 2019), proposing a Global encoder and Slot-Attentive decoders (G-SAT). The G-SAT model uses an RNN to encode the user input and slot-specific feed-forward networks to represent the slot-values.

Encoders based on pre-trained LM. The use of pre-trained language models, such as BERT (Bidirectional Encoder Representation from Transformers) (Devlin et al., 2019), is meant to increase the DST capacity to capture the semantics of slot and values names. (Lee et al., 2019) proposed a slot-utterance matching belief tracker (SUMBT) using BERT to encode slots, user input, and slot-values. The representations of the slots and of the user input are combined using multi-head attention (Vaswani et al., 2017) to obtain the input representation of the model, and then compared with the slot-value representation to obtain the probability.

5 Dynamic Ontology DST Models

The models discussed in Section 4 rely on a fixed slot-value set, which is assumed to be available before making the prediction. This is a severe limitation to domains where compiling the slot-value set

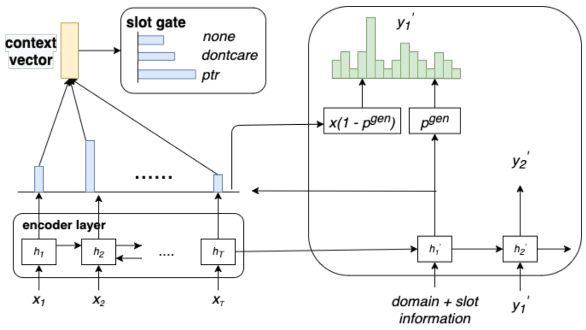


Figure 3: Architecture of the TRADE model using slot-gate and copy mechanism. (Wu et al., 2019).

is costly, or the set of possible slot-values is open (e.g., DEPARTURE_TIME, RESTAURANT_NAME, etc.). For this reason, various studies have focused on developing models that can track slot values even if they are not defined in the ontology. Two major approaches for dynamic ontology models are: i) copy the slot value from the user input to the output; and ii) generate the slot value as the output. Figure 3 presents the schema of a model using the combination of both approaches. One significant difference between static ontology and dynamic ontology models is that while the output vocabulary in the static ontology is limited (i.e., equal to # of slot-values), in a dynamic ontology setting the output vocabulary is much larger.

Copy and pointer networks. Copy mechanism (Gu et al., 2016) and pointer networks (Vinyals et al., 2015) are the main approaches in neural networks to make predictions on the input tokens. They both rely on the attention mechanism (Bahdanau et al., 2015) to obtain scores over the input tokens. (Xu and Hu, 2018) proposed an end-to-end DST architecture based on pointer networks, showing efficient tracking of unseen slot values in a data-driven approach on the DSTC2 dataset. However, since pointer networks can only make predictions on the input tokens, they cannot be directly applied for all slots and require postprocessing of predicted values. (Wu et al., 2019) proposed a Transferable Multi-Domain State Generator *TRADE*, the first generation-based DST that incorporates the copy mechanism with a slot-gate. Figure 3 shows the architecture of the *TRADE* model. *TRADE* is based on an encoder-decoder architecture consisting of a three-way classifier that predicts over probabilities *ptr, none*, and *dontcare*. If the value is not expressed, it is predicted as *none*, if no constraint then *dontcare* and, if the value is expressed in the input, then *ptr* is predicted by the slot-gate. On *ptr* prediction, the corresponding value needs to be decoded by the decoder layer (referred as state generator). The state generator layer is initialized with both the domain and the slot representation, and generates the dialogue state using a recurrent architecture. As all the parameters are shared for all slots and domains, *TRADE* enables the transfer of knowledge from one domain to another, which has opened research directions in zero-shot approaches for DST with promising results.

Categorical and non-categorical slot-values. DST models based on dynamic ontology are supposed to address predictions particularly for non-categorical slots, which admit an open set of values. In this direction (Zhang et al., 2019) proposed a dual-strategy approach that can predict both over a predefined set of slot-values and can generate values based on the input dialogue. If a given slot is labeled as *categorical* (i.e., possible values for the slot are predefined), the output layer predicts a score over the possible slot-values, while, if the slot is labeled as *non-categorical*, the span (i.e., start and end positions) of the value is decoded from the input tokens. (Heck et al., 2020) proposed a triple copy strategy (TripPy) for DST. The slot-values are predicted based on one of the following three scenarios: i) explicitly expressed by the user; ii) expressed by the system and referred to by the user; and iii) expressed in an earlier dialogue turn for another domain-slot. TripPy uses a slot gate to predict the slot status and then uses a copy mechanism to predict the slot-value.

Function-based update. The approaches reported so far for dynamic ontology either use a rule-based update mechanism or they predict the complete dialogue state at each turn from scratch. A function-based update mechanism is proposed in SOM-DST, Selectively Overwriting Memory model (Kim et al., 2020), that tracks the dialogue state in memory and predicts only the dialogue state update. First, one of the four slot operations (i.e., {*CARRYOVER, DELETE, DONTCARE, UPDATE*}) is initially predicted to decide the decoding strategy for the slot. *CARRYOVER* denotes that the slot-value from the previous dialogue state is carried over, *DELETE* denotes that the user retracts the slot-value and *UPDATE* denotes that a new slot-value needs to be predicted and updated to the dialogue state. Then, based on the state update prediction, a dialogue state is decoded.

Schema-guided models. So far, all of DST approaches focus on modeling a given ontology, without considering the portability and flexibility of the model to accommodate other datasets or domains. Though some models, such as *TRADE, SOM-DST, DS-Picklist and TripPy* (Wu et al., 2019; Kim et al., 2020; Zhang et al., 2019; Heck et al., 2020) can make predictions for a new domain, they are typically modeled only for the domains in a specific dataset, and the flexibility of the model to incorporate new domains is not an inherent feature. This is basically due to the different ontology schema used in each dataset, which make them incompatible. In this context, the schema-guided dataset (SGD) (see Section 3.4), puts forth a standard schema to be adopted for all domains. In SGD, a standard schema structure is adopted, slots are classified as either categorical or non-categorical, and each slot includes a brief natural language description. Then, a new dataset needs to follow this schema, which would enable the model to predict dialogue states without any change in the architecture.

Several works exploit the potential of the SGD dataset. (Balaraman and Magnini, 2021) proposed a Domain Aware DST *DA-DST* based on (Rastogi et al., 2020) to effectively predict slot-values specific to each domain. *DA-DST* uses multiple multi-head attention to extract both a domain- and a slot- specific representation from the input, and then combines them to predict the dialogue state. (Chen et al., 2020) use a graph attention network exploiting the slot relations to learn the representation of the ontology schema and the input simultaneously. (Gao et al., 2019) propose a neural reading comprehension approach to DST. Here, for each slot i a question (q_i: *what is the value for slot* i*?*) is formulated and treat the dialogue D_t as a passage. Finally, (Le et al., 2020) propose the first non-auto-regressive DST approach (NADST) to learn the inter-dependencies across slots. This approach allows for a parallel decoding strategy to considerably reduce the latency of the models in-comparison with recurrent architectures.

6 Take-away Points

This section presents take-away points intended to underline both limitations and improvements in different scenarios.

1. Employing various models for each slot limits the models' generalization capability and the ability to learn an effective representation for the input.

2. Parameter sharing among slots (even at the encoder level alone) is effective and improves performance for all slots.

3. When large training data is available, recurrent neural networks are preferred for state-of-the-art performance. In this context, bidirectional architectures are shown to be additive to the models' performance in specific datasets.

4. The latency in recurrent architectures is an issue if used for both encoder and decoder. Recurrent networks process the input one time-step at a time, and employing multiple such networks increases the time required for prediction.

5. The attention-based copying mechanism is an effective approach to make predictions on the user input as slot-values. This approach is used in most of the state-of-the-art models, with some variations.

6. For low-resource domains using pre-trained language models as encoders drastically improves the performance.

7. Statistical update functions are shown to outperform rule-based update functions.

8. When the scalability of the domain and the models flexibility is an issue, adopting the schema-based approach enables the model to incorporate any change in schema. This also enables transfer learning including zero-shot (discussed in Section 7.1).

9. The majority of recent DST models rely on pre-trained language models to encode the model inputs (Heck et al., 2020), which leads to learning better representations and higher performance.

Appendix A provides additional details of the models discussed in this survey.

7 DST Challenges and Future Directions

The addition of new slots and new domains is inevitable in real-world conversational applications when a dialogue system is deployed (Rastogi et al.,

Model	DSTC2	WoZ2.0	MultiWoZ (version)	SGD
Word-based DST (Henderson et al., 2014c)	0.691	-	-	-
Scalable Multi-domain DST (Rastogi et al., 2017)	0.703	-	-	-
Pointer (Xu and Hu, 2018)	0.721	-	-	-
Multi-domain DST (Mrkšić et al., 2015)	0.750	-	-	-
NBT (Mrkšić et al., 2017a)	0.734	0.842	-	-
BERT-DST (Chao and Lane, 2019)	0.693	0.877	-	-
GLAD (Zhong et al., 2018)	0.745	0.881	0.356 (1.0)	-
StateNet (Ren et al., 2018)	**0.755**	0.889	-	-
CNN-Delex (Wen et al., 2017)	-	0.837	-	-
FS-NBT (Mrkšić and Vulić, 2018)	-	0.848	-	-
GCE (Nouri and Hosseini-Asl, 2018)	-	0.885	0.362 (2.0)	-
GSAT (Balaraman and Magnini, 2019)	-	0.887	-	-
DST Reader (single) (Gao et al., 2019)	-	-	0.364 (2.1)	-
TRADE (Wu et al., 2019)	-	-	0.456 (2.1)	-
SUMBT (Lee et al., 2019)	-	0.910	0.466 (2.0)	-
NARDST (Le et al., 2020)	-	-	0.490 (2.1)	-
SOM-DST (Kim et al., 2020)	-	-	0.525 (2.1)	-
DS-Picklist (Zhang et al., 2019)	-	-	0.533 (2.1)	-
MinTL (Lin et al., 2020)	-	-	0.536 (2.1)	
SST (Chen et al., 2020)	-	-	0.552 (2.1)	-
TripPy (Heck et al., 2020)	-	**0.927**	**0.553** (2.1)	-
SGD-Baseline (Rastogi et al., 2020)	-	0.810	0.434 (2.1)	0.254
DA-DST (Balaraman and Magnini, 2021)	-	0.899	0.454 (2.1)	**0.310**

Table 2: Performance (joint goal accuracy) of DST systems on available datasets as reported in respective papers.

2020). Hence, approaches to train models with limited or no training data are much required and it is a challenge in DST to exploit techniques such as few and zero shot learning and data augmentation.

7.1 Few-shot and Zero-shot Models

Initial DST datasets were domain specific and models actually focused on effectively tracking dialogue states defined for those domains (see section 4 and 5). However, the recently published multi-domain datasets and the progress in the field of NLP, have driven the DST community to propose more advanced models that can track multiple domains and even are flexible to be adapted to new domains that are not predefined in the dataset (Mrkšić et al., 2015; Ramadan et al., 2018; Rastogi et al., 2017; Zhong et al., 2018; Nouri and Hosseini-Asl, 2018).

TRADE (see section 5) was the first model investigating zero-shot and few-shot learning approaches on the MultiWoZ dataset, showing promising results on multiple domains. *TRADE* relied on the parameter sharing across all domains and slots to improve performance for low resource domains.

To effectively represent new domains and low resource domains, pre-trained language models were used to encode the user input representation and domain/slot representations (Lee et al., 2019; Kim et al., 2020; Heck et al., 2020; Rastogi et al., 2020; Balaraman and Magnini, 2021). In addition, the schema guided dataset enabled models to be able to predict dialogue states for any domains that adopt the proposed schema, paving the way for further progress in zero-shot learning approaches for DST (Rastogi et al., 2020; Balaraman and Magnini, 2021; Gao et al., 2019).

Finally, (Lin et al., 2020) used the pre-trained T5 (Raffel et al., 2020) and BART (Lewis et al., 2020) language model, and proposed a minimalist transfer learning approach called MinTL. Unlike other models that predict the dialogue state, MinTL generates the change in the dialogue state as a Levenshtein belief state. This unique approach showed more robust results in low resource domains.

7.2 Data Augmentation and Data-efficient Models

The high cost of data acquisition for annotated dialogues has pushed researchers to look for alternative options. Among them, data augmentation allows generating additional training from existing data. In addition, the cost of dialogue collection makes models that can learn from a small amount of data highly preferred, and the use of pre-trained language models in DST architecture has shown promising results. However, current models have shown success solely on selected domains, where the dialogue task is straightforward.

Reinforced data augmentation was proposed by (Yin et al., 2020), using a reinforcement learning approach to learn a data augmentation policy. A *generator* that learns how to generate new data, and a *tracker* trained for DST are learned in an alternate manner. The generator is learned using reinforcement learning rewards, and the tracker is then retrained on the data generated by the generator. This approach showed to significantly improve the DST performance. However, it lacks the controllability of the generated data. CoCo (Controllable Counterfactuals - (LI et al., 2021)) is a recent DST that provides control in generating data with specific slot-values in the utterance. This is achieved by training a conditional generation model using an encoder-decoder framework based on the system response, and the turn-level user goal to generate the user utterance. Once learned, the model can generate a new utterance when a new turn-level user goal is input to the model. A filtering approach was also employed to check if all the desired turn-level user goals are present in the generated output, and to choose the one satisfying the user goal.

7.3 Diverse Datasets

Much of the DST progress was achieved after the release of multi-domain datasets, particularly MultiWoZ and SGD. However, these datasets are not sufficient to train deployment-ready models due to various uncertain situations that the models encounter in the real world, such as linguistic variations and uncooperative users. Moreover, almost all datasets are in English (WoZ2.0 alone was translated to German and Italian).

Another important direction for the future is leveraging other conversational datasets that are widely available in many open social media platforms, such as Reddit and Twitter. As these datasets are open-domain and unlabelled, the main challenge is learning a dialogue structure behind these dialogues that can help learning task-oriented dialogues and be data-efficient.

8 Conclusion

We have surveyed a number of recent studies addressing neural-network-based DST and have discussed both the task and the major datasets available to the research community. We grouped models according to their capacity to make dialogue state predictions either with respect to a static ontology (i.e., a fixed set of dialogue states) or with respect to a dynamic ontology (i.e., an open set of dialogue states). We also reported about DST models' progress towards modeling trackers that perform few-shot and zero-shot learning to accommodate new domains, this way opening multiple opportunities both in research and industry.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

V. Balaraman and B. Magnini. 2019. Scalable neural dialogue state tracking. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 830–837.

V. Balaraman and B. Magnini. 2021. Domain-aware dialogue state tracker for multi-domain dialogue systems. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:866–873.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Guan-Lin Chao and Ian Lane. 2019. BERT-DST: Scalable End-to-End Dialogue State Tracking with Bidirectional Encoder Representations from Transformer. In *Proc. Interspeech 2019*, pages 1468–1472.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explor. Newsl.*, 19(2):25–35.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7521–7528. AAAI Press.

Jianpeng Cheng, Devang Agrawal, Hector Martinez Alonso, Shruti Bhargava, Joris Driesen, Federico Flego, Dain Kaplan, Dimitri Kartsaklis, Lin Li, Dhivya Piraviperumal, et al. 2020. Conversational semantic parsing for dialog state tracking. *arXiv preprint arXiv:2010.12770*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 422–428.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273, Stockholm, Sweden. Association for Computational Linguistics.

Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis lectures on human language technologies*, 10(1):1–309.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. Trippy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2020, 1st virtual meeting, July 1-3, 2020*, pages 35–44. Association for Computational Linguistics.

M. Henderson, B. Thomson, and S. Young. 2014. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 360–365.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014a. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014b. The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 324–329.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014c. Word-based dialog state tracking with recurrent neural networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Hung Le, Richard Socher, and Steven C.H. Hoi. 2020. Non-autoregressive dialog state tracking. In *International Conference on Learning Representations*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

SHIYANG LI, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2021. Coco: Controllable counterfactuals for evaluating dialogue state trackers. In *International Conference on Learning Representations*.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. Mintl: Minimalist transfer learning for task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bing Liu, Gokhan Tür, Dilek Hakkani-Tür, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2060–2069, New Orleans, Louisiana. Association for Computational Linguistics.

Samuel Louvan and Bernardo Magnini. 2020. Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 480–496, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2015. Multi-domain dialog state tracking using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 794–799, Beijing, China. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017a. 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.

Nikola Mrkšić and Ivan Vulić. 2018. Fully statistical neural belief tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 108–113, Melbourne, Australia. Association for Computational Linguistics.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017b. Semantic specialization of distributional word vector spaces using monolingual and cross-lingual constraints. *Transactions of the Association for Computational Linguistics*, 5:309–324.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking. In *NeurIPS 2018, 2nd Conversational AI workshop*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437, Melbourne, Australia. Association for Computational Linguistics.

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

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8689–8696.

Liliang Ren, Kaige Xie, Lu Chen, and Kai Yu. 2018. Towards universal dialogue state tracking. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2780–2786, Brussels, Belgium. Association for Computational Linguistics.

Pararth Shah, Dilek Hakkani-Tür, Bing Liu, and Gokhan Tür. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51, New Orleans - Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France. Association for Computational Linguistics.

Jason D. Williams, Antoine Raux, and Matthew Henderson. 2016a. The dialog state tracking challenge series: A review. *Dialogue Discourse*, 7(3):4–33.

Jason D. Williams, Antoine Raux, and Matthew Henderson. 2016b. The dialog state tracking challenge series: A review. *Dialogue Discourse*, 7(3):4–33.

Jason D. Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech Language*, 21(2):393 – 422.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

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.

Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dialog state tracking with reinforced data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9474–9481.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S. Yu, Richard Socher, and Caiming Xiong. 2019. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. *CoRR*, abs/1910.03544.

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.

A Appendix

Model	Values	Slots	Schema	Update
Word-based DST (Henderson et al., 2014c)	Closed	Closed	Fixed	Function
Multi-domain DST (Mrkšić et al., 2015)	Closed	Closed	Fixed	Function
FS-NBT (Mrkšić and Vulić, 2018)	Closed	Closed	Fixed	Function
Scalable Multi-domain DST (Rastogi et al., 2017)	Closed	Closed	Fixed	Rules
CNN-Delex (Wen et al., 2017)	Closed	Closed	Fixed	Rules
NBT (Mrkšić et al., 2017a)	Closed	Closed	Fixed	Rules
StateNet (Ren et al., 2018)	Closed	Open*	Fixed	Rules
Pointer (Xu and Hu, 2018)	Open	Closed	Fixed	Rules
GLAD (Zhong et al., 2018)	Closed	Closed	Fixed	Rules
GCE (Nouri and Hosseini-Asl, 2018)	Closed	Open	Fixed	Rules
GSAT (Balaraman and Magnini, 2019)	Closed	Closed	Fixed	Rules
BERT-DST (Chao and Lane, 2019)	Open	Closed	Fixed	Rules
TRADE (Wu et al., 2019)	Open	Open*	Dynamic	None
DS-Picklist (Zhang et al., 2019)	Closed	Open	Fixed	None
SUMBT (Lee et al., 2019)	Closed	Open	Fixed	Function
SST (Chen et al., 2020)	Closed	Open*	Fixed	Function
SGD-Baseline (Rastogi et al., 2020)	Open	Open	Dynamic	Rules
DA-DST (Balaraman and Magnini, 2021)	Open	Open	Dynamic	Rules
SOM-DST (Kim et al., 2020)	Open	Open	Dynamic	Function
TripPy (Heck et al., 2020)	Open	Open	Dynamic	Function
MinTL (Lin et al., 2020)	Open	Open	Dynamic	Function
Nerual Reading (Gao et al., 2019)	Open	Open	Dynamic	Function
NARDST (Le et al., 2020)	Open	Open	Dynamic	None

Table 3: Tracking approach of implemented by various DST models. $*$ denotes the requirement of a pre-trained embedding