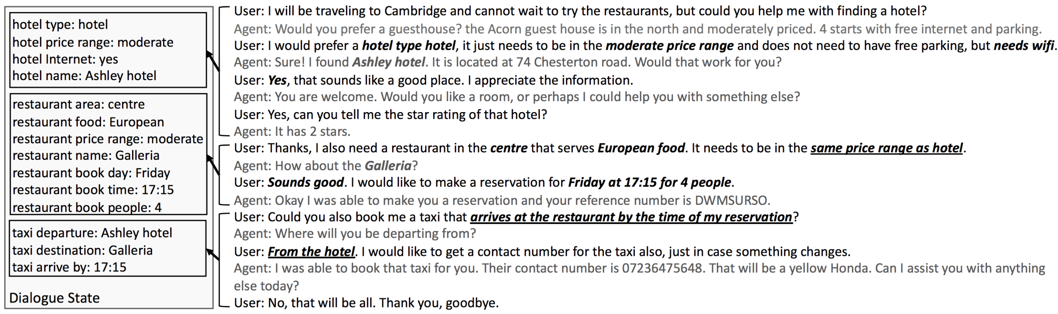

Figure 1: The right-hand side shows a dialogue which involves 3 domains, and the left-hand side shows its dialogue state in the end. Bold text indicate mentions and paraphrases of slot values. Underlined text indicates scenarios where multi-turn inference is required.

a 5 star hotel, then the user is likely looking for an expensive rather than a cheap restaurant. As we will show later, such relationships between domains and slots help improve model performance.

To tackle these challenges, we propose DSTQA (*Dialogue State Tracking via Question Answering)*, a new multi-domain DST model inspired by recently developed reading comprehension and question answering models. Our model reads dialogue contexts to answer a series of questions that asks for the value of a (domain, slot) pair. Specifically, we construct two types of questions: 1) multiple choice questions for (domain, slot) pairs with a limited number of value options and 2) span prediction questions, of which the answers are spans in the contexts, designed for (domain, slot) pairs that have a large or infinite number of value options. Finally, we represent (domain, slot) pairs as a dynamically-evolving knowledge graph with respect to the dialogue context, and utilize this graph to drive improved model performance. Our contributions are as follows: (1) we propose to model multi-domain DST as a question answering problem such that tracking new domains, new slots and new values is simply constructing new questions, (2) we propose using a bidirectional attention (Seo et al., 2017) based model for multi-domain dialogue state tracking, and (3) we extend our algorithm with a dynamically-evolving knowledge graph to further exploit the structure between domains and slots.

2 Problem Formulation

In a multi-domain dialogue state tracking problem, there are M domains $D = \{d_1, d_2, ..., d_M\}$. For example, in MultiWOZ 2.0/2.1 datasets, there are 7 domains: *restaurant*, *hotel*, *train*, *attraction*, *taxi*, *hospital*, and *police*. Each domain $d \in D$ has N^d slots $S^d = \{s_1^d, s_2^d, ..., s_{N^d}^d\}$, and each slot $s \in S^d$ has K^s possible values $V^s = \{v_1^s, v_2^s, ..., v_{K^s}^s\}$. For example, the *restaurant* domain has a slot named *price range*, and the possible values are `cheap`, `moderate`, and `expensive`. Some slots do not have pre-defined values, that is, V^s is missing in the domain ontology. For example, the *taxi* domain has a slot named *leave time*, but it is a poor choice to enumerate all the possible leave times the user may request as the size of V^s will be very large. Meanwhile, the domain ontology can also change over time. Formally, we represent a dialogue X as $X = \{U_1^a, U_1^u, U_2^a, U_2^u, ..., U_T^a, U_T^u\}$, where U_t^a is the agent utterance in turn t and U_t^u is the user utterance in turn t. Each turn t is associated with a dialogue state y_t. A dialogue state y_t is a set of (domain, slot, value) tuples. Each tuple represents that, up to the current turn t, a slot $s \in S^d$ of domain $d \in D$, which takes the value $v \in V^s$ has been provided by the user. Accordingly, y_t's are targets that the model needs to predict.

3 Multi-domain Dialogue State Tracking via Question Answering (DSTQA)

We model multi-domain DST as a question answering problem and use machine reading methods to provide answers. To predict the dialogue state at turn t, the model observes the context C_t, which is the concatenation of $\{U_1^a, U_1^u, ..., U_t^a, U_t^u\}$. The context is read by the model to answer the questions defined as follows. First, for each domain $d \in D$ and each slot $s \in S^d$ where there exists a pre-defined value set V^s, we construct a question $Q_{d,s} = \{d, s, V^s, \texttt{not mentioned}, \texttt{don't care}\}$. That is, a question is a set of words or phrases which includes a domain name, a slot name, a list of all possible values, and two special values `not mentioned` and `don't`