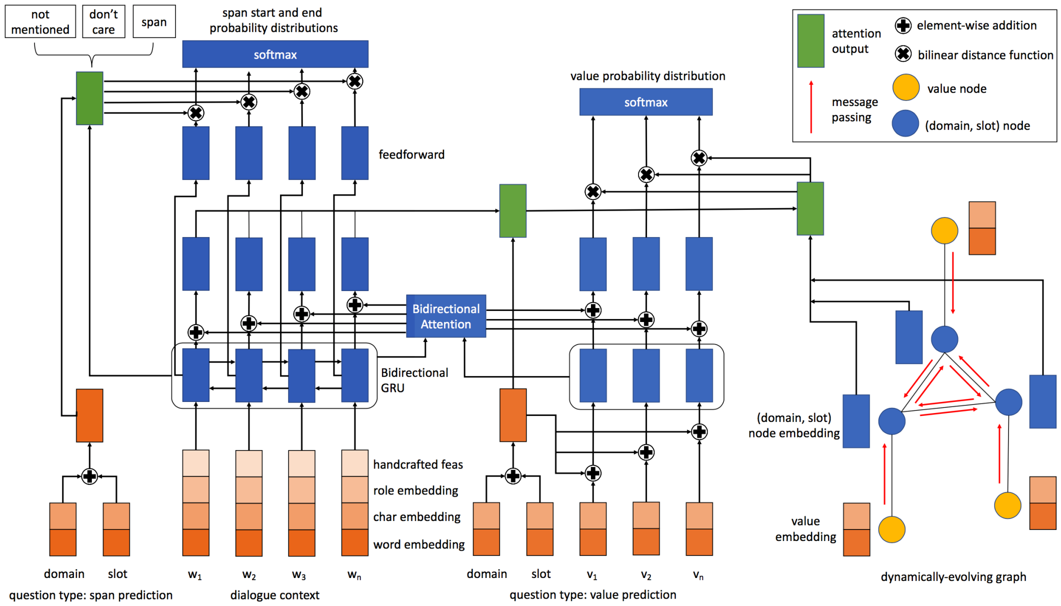

Figure 2: DSTQA model architecture. When the question type is value prediction, a bidirectional attention layer is applied to the dialogue context and the question, and a graph embedding is injected to the output of the bidirectional attention layer. When the question type is span prediction, the question is used to attend over the dialogue context to predict span start and end positions.

2. Context Encoding Layer: We apply a bidirectional GRU to encode the context C_t. Denoting the i-th word in the context C_t by w_i, then the input to the bidirectional GRU at time step i is the concatenation of the following three vectors: 1) w_i's word embeddings, $W^c_{i,:}$, 2) the corresponding role embedding, and 3) exact match features. There are two role embeddings: the agent role embedding $e_a \in \mathbb{R}^r$ and the user role embedding $e_u \in \mathbb{R}^r$ where both are trainable. Exact match features are binary indicator features where for each (domain, slot) pair, we search for occurrences of its values in the context in original and lemmatized forms. Then for each (domain, slot) pair, we use two binary features to indicate whether w_i belongs to an occurrence in either form. The final output of this layer is a matrix $E^c \in \mathbb{R}^{L_c \times D^{\text{biGRU}}}$, where L_c is the number of words in the context C_t and D^{biGRU} is the dimension of bidirectional GRU's hidden states (includes both forward and backward hidden states). In our experiments, we set D^{biGRU} equals to D^w.

3. Question-Context Bidirectional Attention Layer: Inspired by Seo et al. (2017), we apply a bidirectional attention layer which computes attention in two directions: from context C_t to question $Q_{d,s}$, and from question $Q_{d,s}$ to context C_t. To do so, we first define an attention function $\mathbb{R}^{m*n} \times \mathbb{R}^n \to \mathbb{R}^m$ that will be used frequently in the following sections. The inputs to the function are a key matrix $K \in \mathbb{R}^{m*n}$ and a query vector $q \in \mathbb{R}^n$. The function calculates the attention score of q over each row of K. Let $O \in \mathbb{R}^{m*n}$ be a matrix which is q repeated by m times, that is, $O_{:,j} = q$ for all j. Then, the attention function is defined as:

$$\text{Att}_\beta(K, q) = \text{Softmax}([K; O^\top; K \odot O^\top] \cdot \beta)$$

Where $\beta \in \mathbb{R}^{3n}$ are learned model parameters, $\odot$ is the element-wise multiplication operator, and $[;]$ is matrix row concatenation operator. We use subscript of β, β_i, to indicate different instantiations of the attention function.

The attention score of a context word w_i to values in $Q_{d,s}$ is given by $\alpha^v_i = \text{Att}_{\beta_1}(W^q, E^c_{i,:}) \in \mathbb{R}^{L_{\bar{v}}}$, and the attention score of a value v_j to context words in C_t is given by $\alpha^w_j = \text{Att}_{\beta_1}(E^c, W^q_{j,:}) \in \mathbb{R}^{L_c}$. β_1 is shared between these two attention functions. Then, the question-dependent embedding of context word w_i is $B^{QD}_i = W^{q\top} \cdot \alpha^v_i$ and can be viewed as the representation of w_i in the vector space defined by the question $Q_{d,s}$. Similarly, the context-dependent embedding for value v_j is $B^{CD}_j = E^{c\top} \cdot \alpha^w_j$ and can be viewed as the representation of v_j in the vector space defined by the context C_t. The final context embedding is $B^c = E^c + B^{QD} \in \mathbb{R}^{L_c \times D^w}$ and the final question embedding is $B^q = B^{CD} + W^q \in \mathbb{R}^{L_{\bar{v}} \times D^w}$.