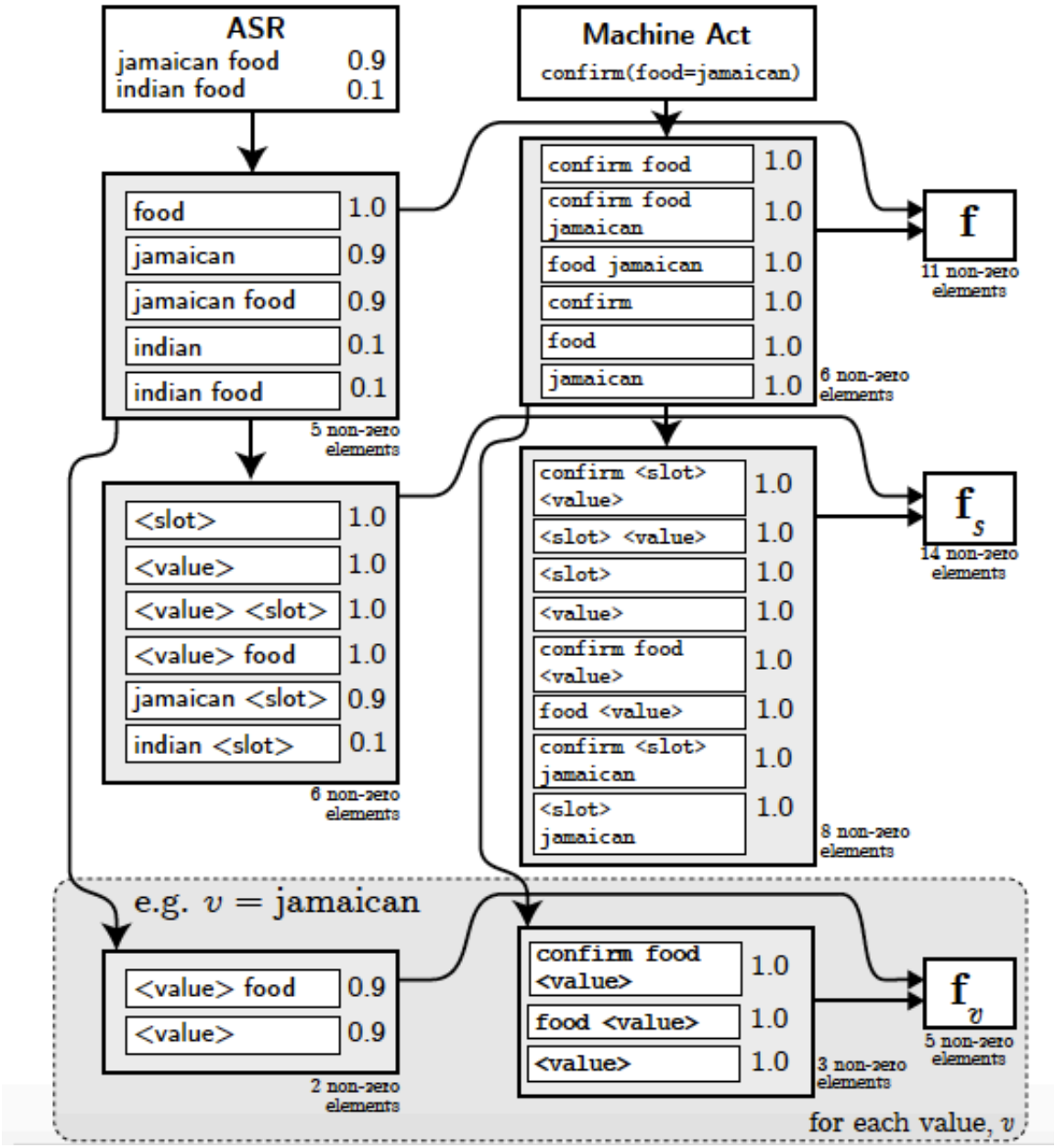
This paper gives an RNN architecture to do dialog state tracking. At each step of the multi-turn dialog, the model first extracts feature from the ASR input and preceding dialog act, sends this feature vector to the RNN module, computes a distribution over requested slots and values respectively and updates the RNN state.

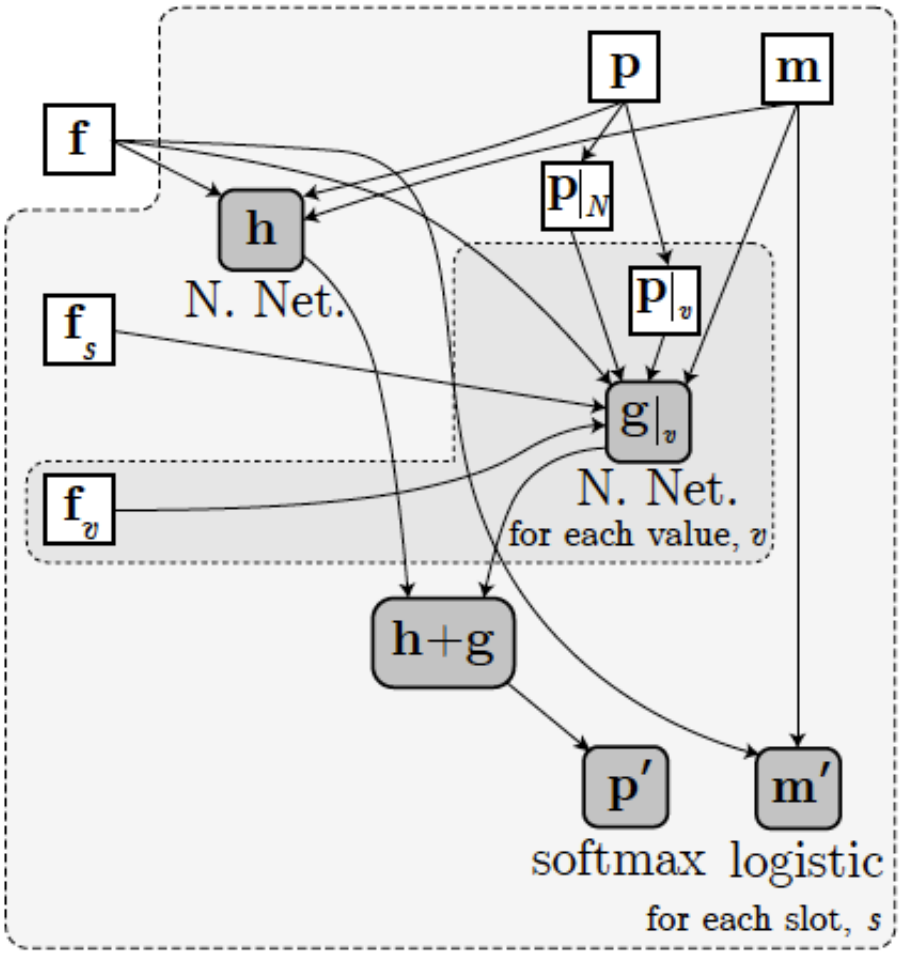
Feature Representation

This model utilizes n-gram features of the ASR N-best list. Specifically it extracts all n-grams from the N-best list and weight them according to the ASR score. N-grams of the last dialog act are also extracted. After concatenating these two vectors, we get the feature for values f. Then for each extracted n-gram, replace at least one word to its generic type to get the *slot feature* f_s (for example *indian food* could generate *food*, *indian* and). This process is illustrated in the figure.



Dialog State Tracking

Two similar RNNs are defined to handle **slots** and **values** respectively. Let **p** denotes the last probability distribution (either for slots or values) and **m** the current RNN state. First it calculates a hidden vector **h** as $\mathbf{h} = MLP(\mathbf{f} + \mathbf{m} + \mathbf{p})$. Then it obtains another hidden vector $\mathbf{g} = \phi(\mathbf{f}, \mathbf{f}_s, \mathbf{p}, \mathbf{m})$, where ϕ is a mapping function defined in the figure below.



Finally it calculates the distribution $\mathbf{p}' = \Phi(\mathbf{g}, \mathbf{h})$, where Φ is also a function. The state of RNNs is updated as $\mathbf{m}' = \sigma(W_1\mathbf{f} + W_2\mathbf{m})$.