

This paper introduces a neural method for the NLG module of a dialog system.

Formally we want to model the probability $p(\mathbf{w}|\mathbf{d}; \theta)$, where d is the semantic input (for example the dialog act together with slots and values $action(s(v))$). We use a RNN as the language model to provide the probability distribution so that we can choose $argmax_w p(w|\mathbf{d}; \theta) = \prod_{i=1}^n p(w_i|w_{1:i-1}, \mathbf{d}; \theta)$.

As stated in the paper, if we simply map the semantic input to a vector d as input and send it to LSTM, due to the problem of gradient vanishing its effect will gradually diminish. If feeding d at each step, we'll face the problem of **duplication** of the semantic labels.

Following the thoughts of *memory* and *forget* in LSTM, the authors designed another gate for the semantic part. That gate takes the last state and current output as input, renews gate state and changes the weight paid on semantic part, formally written as:

$$\begin{aligned} r_t &= \sigma(W_{wr}w_t + W_{hr}h_{t-1}) \\ d_t &= r_t \circ d_{t-1} \end{aligned}$$

And the cell state of LSTM is updated as:

$$c_t = f_t \circ c_{t-1} + i_t \circ \hat{c}_t + \tanh(W_{dc}d_t)$$

The computation flow in the cell is illustrated in this figure:

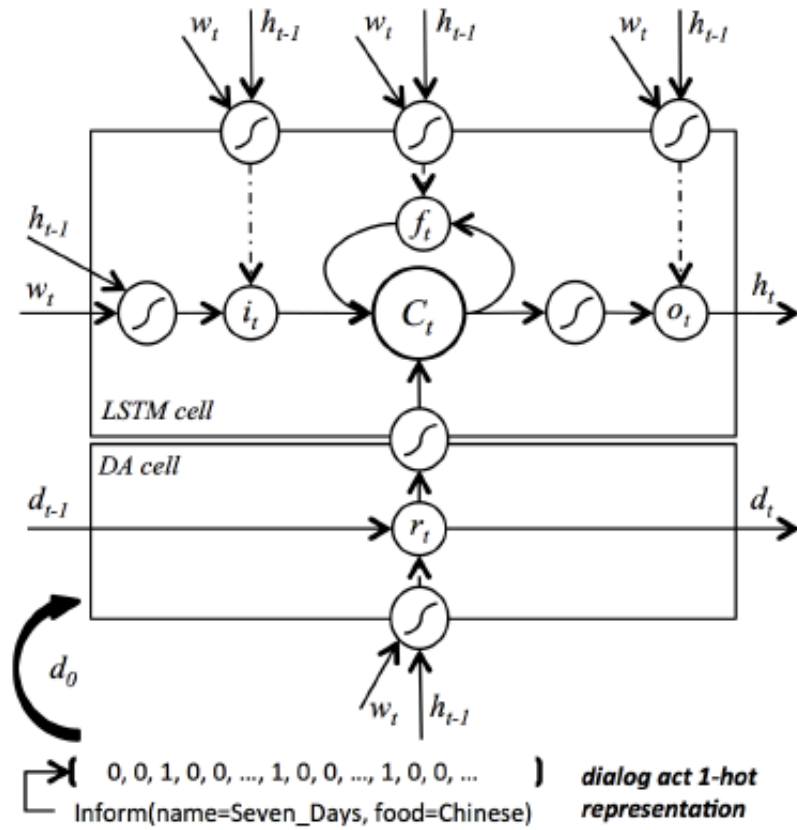


Figure 1: Semantic Controlled LSTM cell proposed in this paper. The upper part is a traditional LSTM cell in charge of surface realisation, while the lower part is a sentence planning cell based on a sigmoid control gate and a dialogue act (DA).

More improvements are tried in this paper, which we leave out here. For full paper see [this link](#).