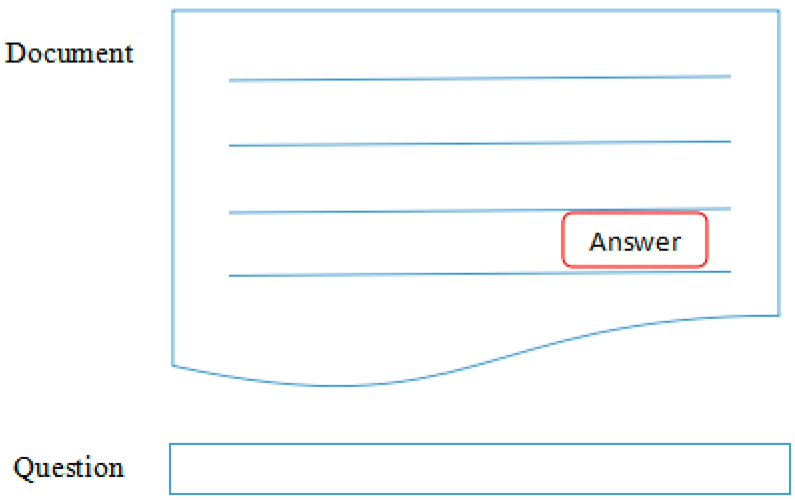
DCN+: Mixed Objective and Deep Residual Coattention for Question Answering

Caiming Xiong, Victor Zhong, Richard Socher
Salesforce Research

Task



Current Model: predicting the start index and end index of answer

Problem

There is a disconnect between optimization and evaluation.

ex.
Sentence: *Some believe that the Golden State Warriors team of 2017 is one of the greatest teams in NBA history.*
Question: *which team is considered to be one of the greatest teams in NBA history?*
A Ground Truth: *The Golden State Warriors team of 2017*

The answer "Warriors" is no better than answer "history".

Contributions

1. It propose a mixed objective that combines traditional CE loss with RL(reward is word overlap).
2. It extend the Dynamic Coattention Network(DCN) with a deep residual coattention encoder.

Shortcut Connections

High-Way Network

$y = H(x) \odot T(x) - x \odot C(x)$

Here, T is the transform gate and C is the carry gate.Usually, C = 1 - T. So

$y = H(x) \odot T(x) + x \odot (1 - T(x))$

$T(x) = \sigma(Wx + b)$

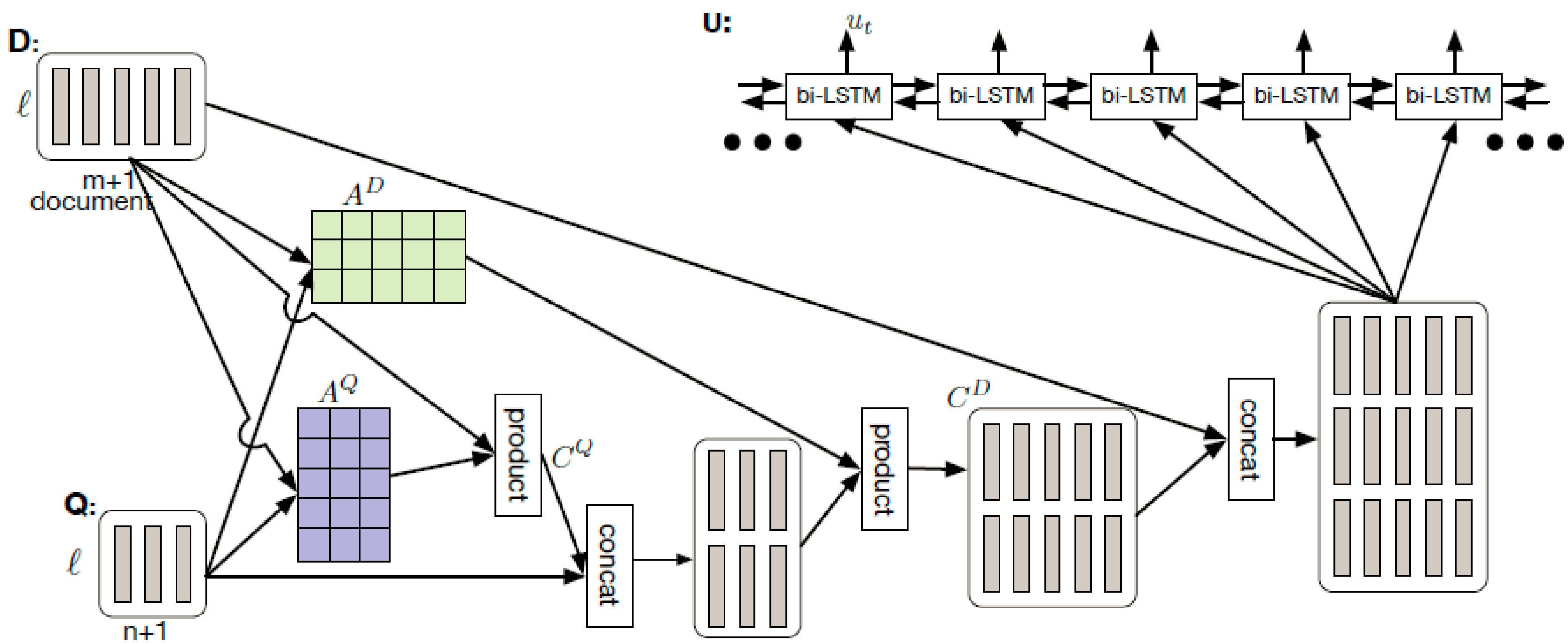
Residual Network

Residual Netword is a specially case of high-way network. T and C is 1

$y = H(x) + x$

Both of them can relief the gradient vanishing problem.

Baseline DCN



DCN+ Encoder

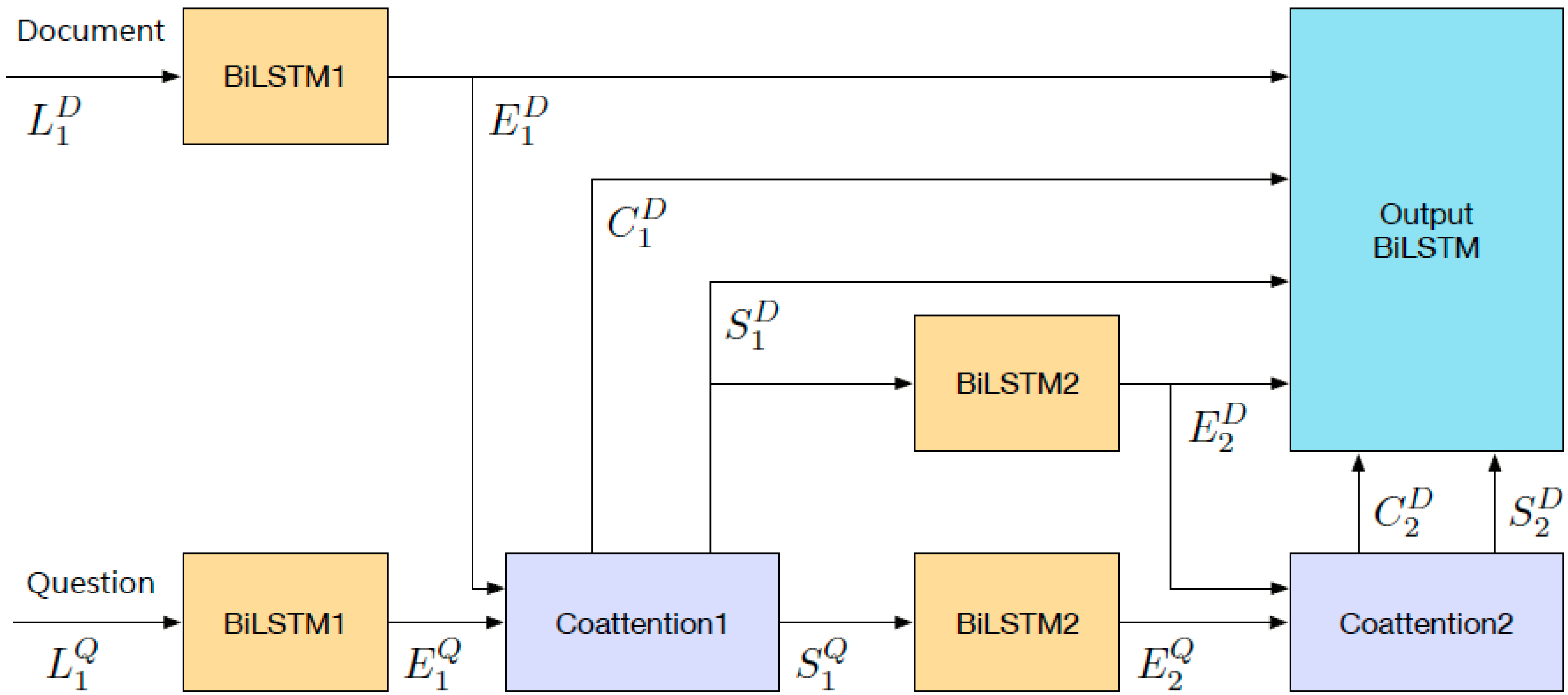
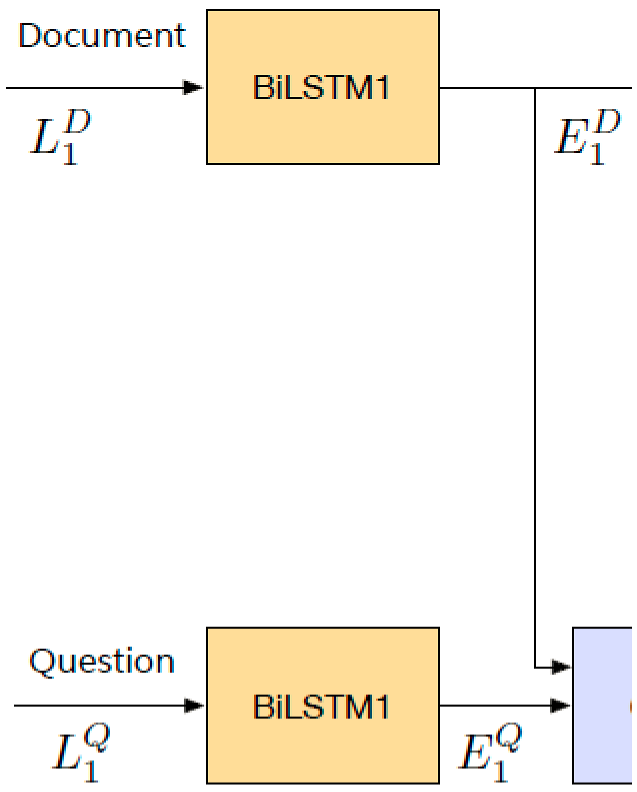


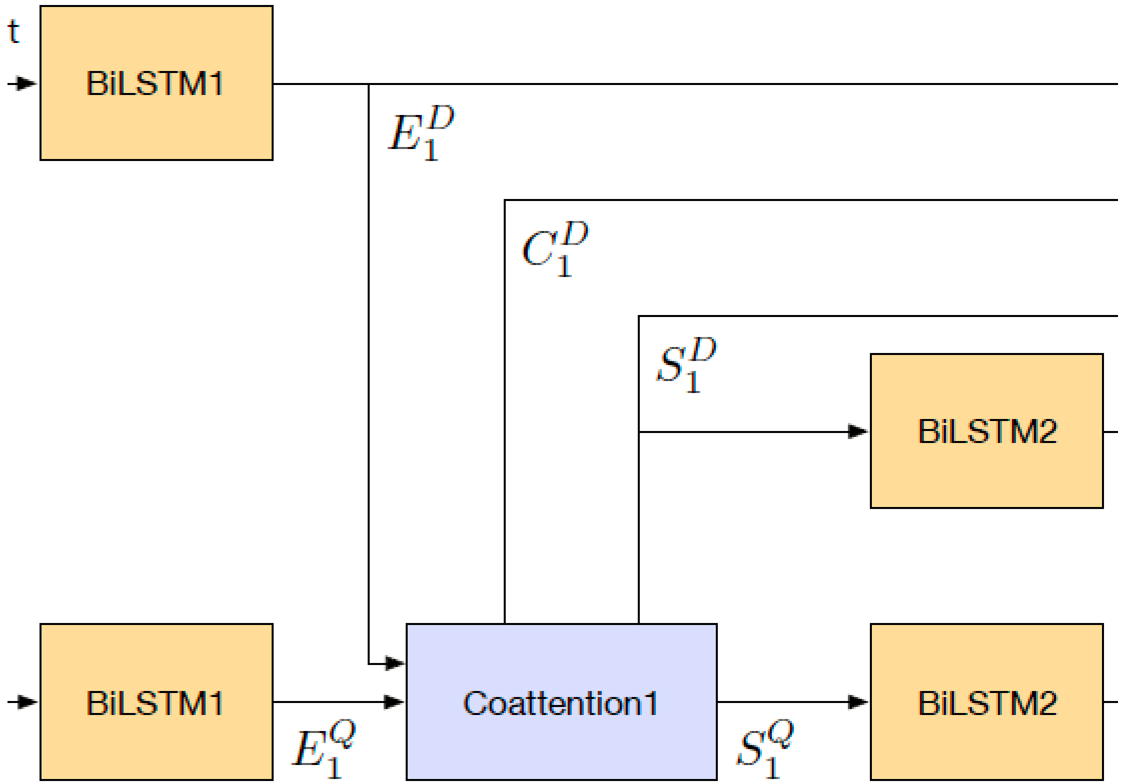
Figure 1: Deep residual coattention encoder.

DCN+ Encoder



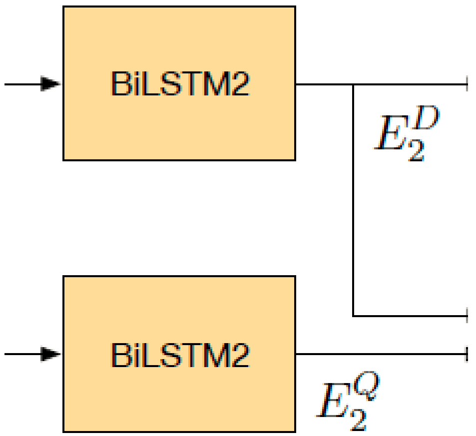
$$E_1^D = biLSTM_1(L^D) \in R^{h\times(m+1)}$$
$$E_1^Q = tanh(W biLSTM_1(L^Q) + b) \in R^{h\times(n+1)}$$

DCN+ Encoder



$$A = (E_1^D)^T E_1^Q \in R^{(m+1)\times(n+1)}$$
$$S_1^D = E_1^Q softmax(A^T) \in R^{h\times(m+1)}$$ *doc-to-que*
$$S_1^Q = E_1^D softmax(A) \in R^{h\times(n+1)}$$ *que-to-doc*
$$C_1^D = S_1^Q softmax(A^T) \in R^{h\times m}$$

DCN+ Encoder



Encoding the summaries using another biLSTM.

$$E_2^D = biLSTM_2(S_1^D) \in R^{2h \times m}$$

$$E_2^Q = biLSTM_2(S_1^Q) \in R^{2h \times n}$$

DCN+ Encoder

Computing the second coattention layer in a similar fashion.Namely,

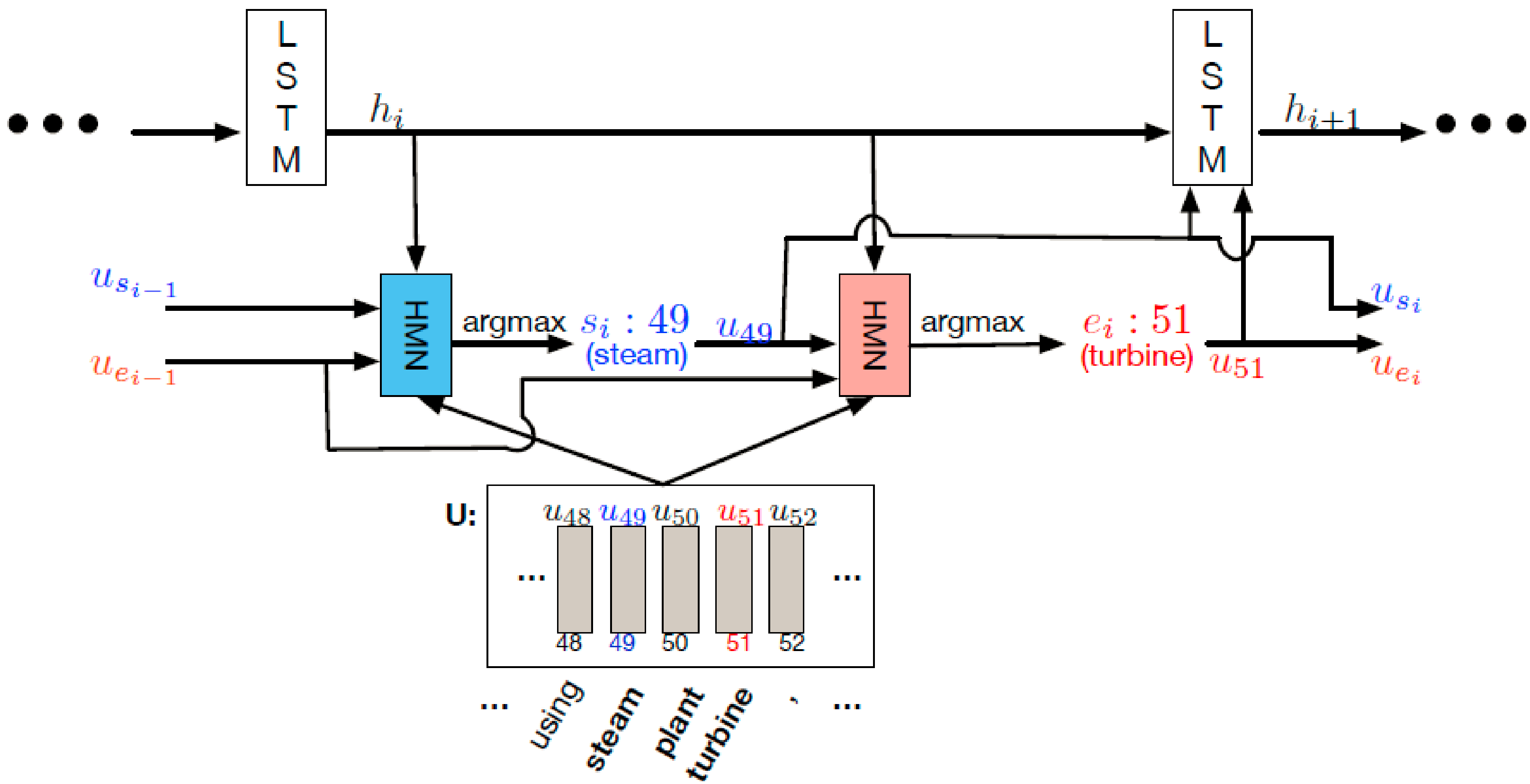
$coattn_1(E_1^D, E_1^Q) -> S_1^D, S_1^Q, C_1^D$

$coattn_2(E_2^D, E_2^Q) -> S_2^D, S_2^Q, C_2^D$

The Output of encoder is obtained as

$U = biLSTM(concat(E_1^D; E_2^D; S_1^D; S_2^D; C_1^D; C_2^D))$

DCN Dynamic Decoder



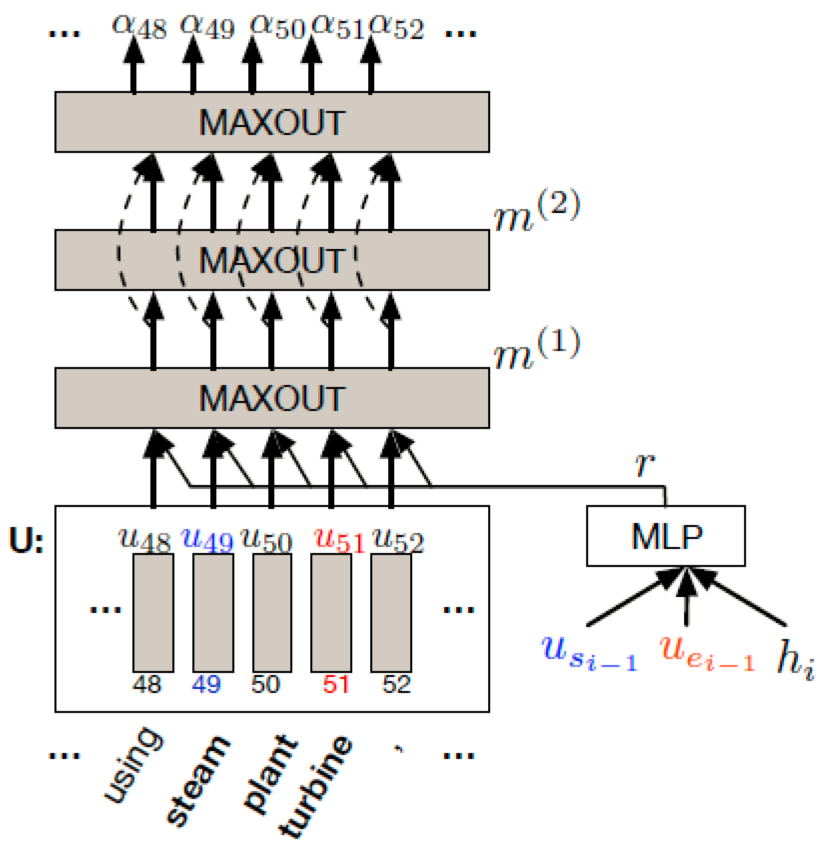
$$h_i = LSTM_{dec}(h_{i-1}, [u_{s_{i-1}}; u_{e_{i-1}}])$$

$$s_i = argmax_t(\alpha_1, ..., \alpha_m)$$

$$e_i = argmax_t(\beta_1, ..., \beta_m)$$

$$\alpha_t = HMN_{start}(u_t, h_i, u_{s_{i-1}}, u_{e_{i-1}})$$

DCN Dynamic Decoder

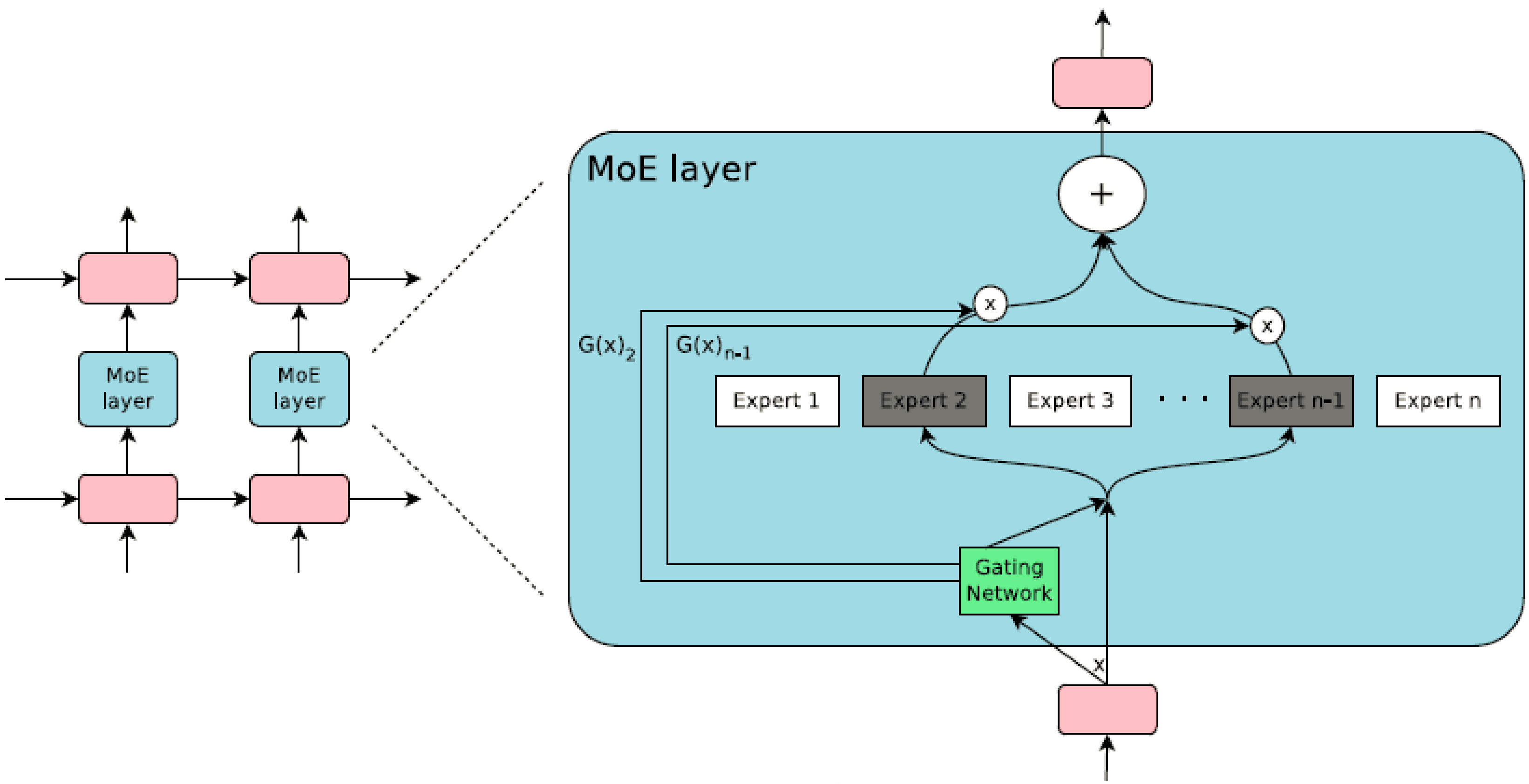


$$
\begin{aligned}
\mathrm{HMN}\left(u_t, h_i, u_{s_{i-1}}, u_{e_{i-1}}\right) &= \max\left(W^{(3)}\left[m_t^{(1)}; m_t^{(2)}\right] + b^{(3)}\right) \\
r &= \tanh\left(W^{(D)}\left[h_i; u_{s_{i-1}}; u_{e_{i-1}}\right]\right) \\
m_t^{(1)} &= \max\left(W^{(1)}\left[u_t; r\right] + b^{(1)}\right) \\
m_t^{(2)} &= \max\left(W^{(2)} m_t^{(1)} + b^{(2)}\right)
\end{aligned}
$$

DCN+ Dynamic Decoder

Swapping the first maxout layer of the highway maxout netword (HMN) with a sparse mixture of experts layer (MoE)(Shazeer et al., 2017)

MoE



Optimization Objective

Cross-Entropy + Reinforcement learning

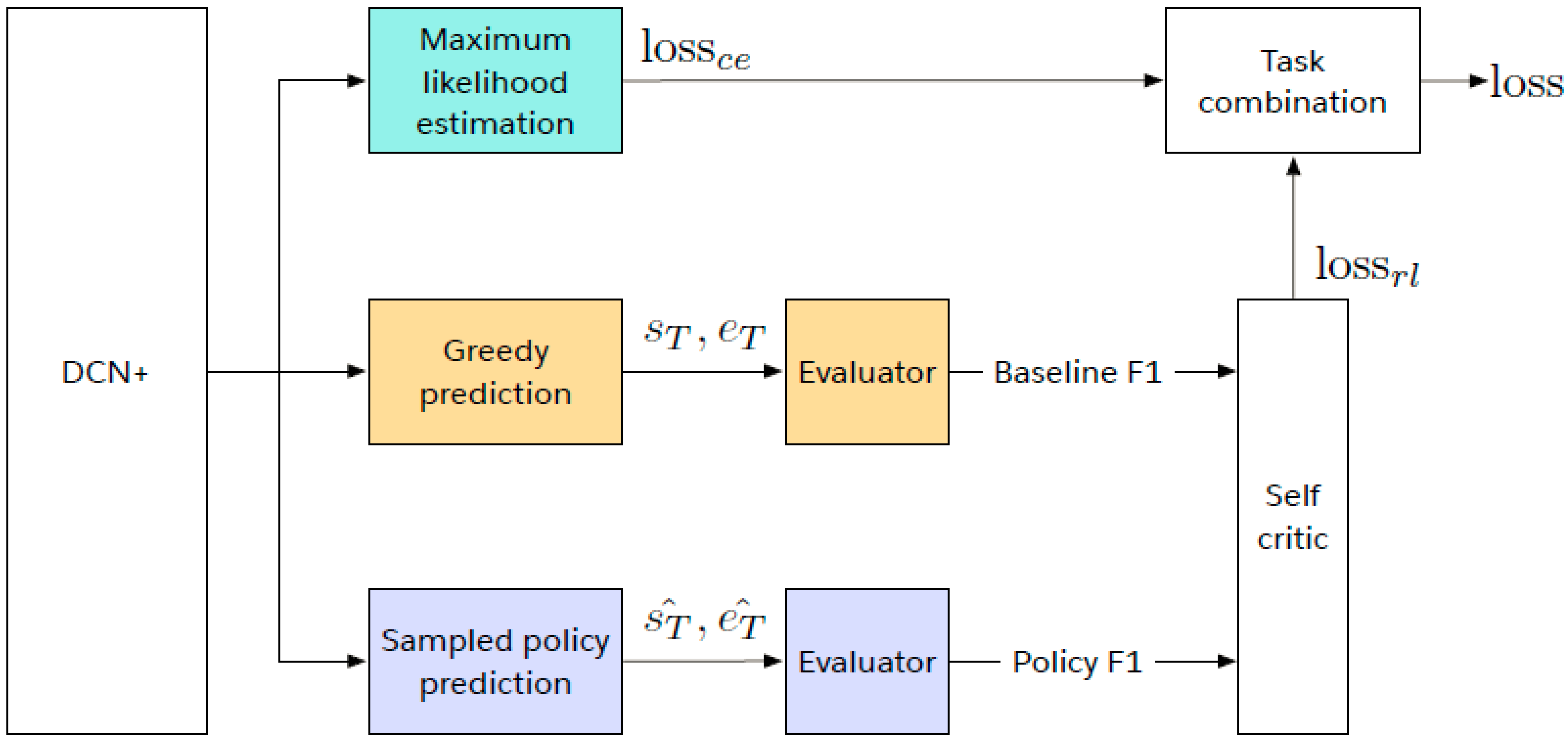


Figure 2: Computation of the mixed objective.

Optimization Objective

Cross-Entropy Objective

$$l_{ce}(\Theta) = -\sum_{t} \left(\log p_t^{\text{start}}\left(s \mid s_{t-1}, e_{t-1}; \Theta\right) + \log p_t^{\text{end}}\left(e \mid s_{t-1}, e_{t-1}; \Theta\right)\right)$$

Reinforcement Learning Objective(F1 is the reward)

$$\begin{aligned} l_{rl}(\Theta) &= -\mathbb{E}_{\hat{\tau} \sim p_\tau}\left[R\left(s, e, \hat{s}_T, \hat{e}_T; \Theta\right)\right] \\ &\approx -\mathbb{E}_{\hat{\tau} \sim p_\tau}\left[F_1\left(\mathrm{ans}\left(\hat{\mathrm{s}}_{\mathrm{T}}, \hat{\mathrm{e}}_{\mathrm{T}}\right), \mathrm{ans}(\mathrm{s}, \mathrm{e})\right) - F_1\left(\mathrm{ans}\left(\mathrm{s}_{\mathrm{T}}, \mathrm{e}_{\mathrm{T}}\right), \mathrm{ans}(\mathrm{s}, \mathrm{e})\right)\right] \end{aligned}$$

The gradient computation of reward function (single Monte-Carlo sample)

$$\begin{aligned} \nabla_\Theta l_{rl}(\Theta) &= -\nabla_\Theta\left(\mathbb{E}_{\hat{\tau} \sim p_\tau}[R]\right) && (14) \\ &= -\mathbb{E}_{\hat{\tau} \sim p_\tau}\left[R \nabla_\Theta \log p_\tau(\tau; \Theta)\right] && (15) \\ &= -\mathbb{E}_{\hat{\tau} \sim p_\tau}\left[R \nabla_\Theta\left(\sum_t^T\left(\log p_t^{\text{start}}\left(\hat{s}_t \mid \hat{s}_{t-1}, \hat{e}_{t-1}; \Theta\right) + \log p_t^{\text{end}}\left(\hat{e}_t \mid \hat{s}_{t-1}, \hat{e}_{t-1}; \Theta\right)\right)\right)\right] \\ &\approx -R \nabla_\Theta\left(\sum_t^T\left(\log p_t^{\text{start}}\left(\hat{s}_t \mid \hat{s}_{t-1}, \hat{e}_{t-1}; \Theta\right) + \log p_t^{\text{end}}\left(\hat{e}_t \mid \hat{s}_{t-1}, \hat{e}_{t-1}; \Theta\right)\right)\right) && (16) \end{aligned}$$

Joint Learning

Combining the two losses using **homoscedastic uncertainty** Kendall et al. (2017) as task-dependent weightings.

$$l = \frac{1}{2\sigma_{ce}^2} l_{ce}\left(\Theta\right) + \frac{1}{2\sigma_{rl}^2} l_{rl}\left(\Theta\right)$$

Here, σ_{ce} and σ_{rl} are learned parameters.

In fact, it is very difficult for policy learning to converge due to the large space of potential answers, documents, and questions if without the cross-entropy loss.

Experiments

Model	Single Model Dev		Single Model Test		Ensemble Test	
	EM	F1	EM	F1	EM	F1
DCN+ (ours)	**74.5%**	**83.1%**	**75.1%**	**83.1%**	**78.9%**	**86.0%**
rnet	72.3%	80.6%	72.3%	80.7%	76.9%	84.0%
DCN w/ CoVe (baseline)	71.3%	79.9%	–	–	–	–
Mnemonic Reader	70.1%	79.6%	69.9%	79.2%	73.7%	81.7%
Document Reader	69.5%	78.8%	70.0%	79.0%	–	–
FastQA	70.3%	78.5%	70.8%	78.9%	–	–
ReasoNet	–	–	69.1%	78.9%	73.4%	81.8%
SEDT	67.9%	77.4%	68.5%	78.0%	73.0%	80.8%
BiDAF	67.7%	77.3%	68.0%	77.3%	73.7%	81.5%
DCN	65.4%	75.6%	66.2%	75.9%	71.6%	80.4%

ps. Context vectors (CoVe) is a kind of embedding feature trained on WMT (McCann et al., 2017).

Experiments

Ablation study

Model	EM	ΔEM	F1	ΔF1
DCN+ (ours)	74.5%	–	83.1%	–
- Deep residual coattention	73.1%	-1.4%	81.5%	-1.6%
- Mixed objective	73.8%	-0.7%	82.1%	-1.0%
- Mixture of experts	74.0%	-0.5%	82.4%	-0.7%
DCN w/ CoVe (baseline)	71.3%	-3.2%	79.9%	-3.2%

Thank you