Lite Unified Modeling for Discriminative Reading Comprehension

Yilin Zhao[1,2], Hai Zhao[1,2,*], Libin Shen[3], Yinggong Zhao[3]
[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3]Leyan Tech, Shanghai, China
zhaoyilin@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn
libin@leyantech.com, ygzhao@leyantech.com

Abstract

As a broad and major category in machine reading comprehension (MRC), the generalized goal of discriminative MRC is answer prediction from the given materials. However, the focuses of various discriminative MRC tasks may be diverse enough: multi-choice MRC requires model to highlight and integrate all potential critical evidence globally; while extractive MRC focuses on higher local boundary preciseness for answer extraction. Among previous works, there lacks a unified design with pertinence for the overall discriminative MRC tasks. To fill in above gap, we propose a lightweight **POS**-Enhanced **I**terative Co-Attention **Net**work (*POI-Net*) as the first attempt of unified modeling with pertinence, to handle diverse discriminative MRC tasks synchronously. Nearly without introducing more parameters, our lite unified design brings model significant improvement with both encoder and decoder components. The evaluation results on four discriminative MRC benchmarks consistently indicate the general effectiveness and applicability of our model, and the code is available at https://github.com/Yilin1111/poi-net.

Multi-choice MRC Example
... *In addition, Lynn's pioneering efforts also provide public educational* ***forums*** *via* ***Green Scenes*** *–* ***a series of three hour events****, each focusing on specific topics teaching Hoosiers how to lead* ***greener lifestyles****. ...*
Q: *What can we learn about* ***Green Scenes****?* A. *It is a scene set in a* ***three-hour film****.* B. *It is* ***a series of events*** *focusing on* ***green life****.* (**Golden**) C. *It is a film set in Central Indiana.* D. *It is a* ***forum*** *focusing on* ***green lifestyle****.*
Extractive MRC Example
... *Early versions were in use by 1851, but the most successful indicator was developed for the high speed engine inventor and manufacturer Charles Porter by Charles Richard and exhibited* ***at London Exhibition in 1862****. ...*
Q: ***Where*** *was the Charles Porter steam engine indicator shown?* Golden Answer: *London Exhibition* Imprecise Answer 1: *London Exhibition* ***in 1862*** Imprecise Answer 2: ***exhibited at*** *London Exhibition*

Table 1: Different focuses of multi-choice MRC task (RACE) and extractive MRC task (SQuAD 2.0). Texts in bold are the critical information or fallibility parts.

1 Introduction

Machine reading comprehension (MRC) as a challenging branch in NLU, has two major categories: generative MRC which emphasizes on answer generation (Kočiský et al., 2018), and discriminative MRC which focuses on answer prediction from given contexts (Baradaran et al., 2020). Among them, discriminative MRC is in great attention of researchers due to its plentiful application scenarios, such as extractive and multi-choice MRC two major subcategories. Given a question with corresponding passage, extractive MRC asks for precise answer span extraction in passage (Joshi et al., 2017; Trischler et al., 2017; Yang et al., 2018), while multi-choice MRC requires suitable answer selection among given candidates (Huang et al., 2019; Khashabi et al., 2018). Except for the only common goal shared by different discriminative MRCs, the focuses of extractive and multi-choice MRC are different to a large extent due to the diversity in the styles of predicted answers: multi-choice MRC usually requires to highlight and integrate all potential critical information among the whole passage; while extractive MRC pays more attention to precise span boundary extraction at local level, since the rough scope of answer span can be located relatively easily, shown in Table 1.

In MRC field, several previous works perform general-purpose language modeling with considerable computing cost at encoding aspect (Devlin et al., 2019; Clark et al., 2020; Zhang et al., 2020c), or splice texts among diverse MRC tasks simply to expand training dataset (Khashabi et al., 2020), without delicate and specialized design for sub-

* Corresponding author. This paper was partially supported by Key Projects of National Natural Science Foundation of China under Grants U1836222 and 61733011.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8682 - 8695
May 22-27, 2022 ©2022 Association for Computational Linguistics

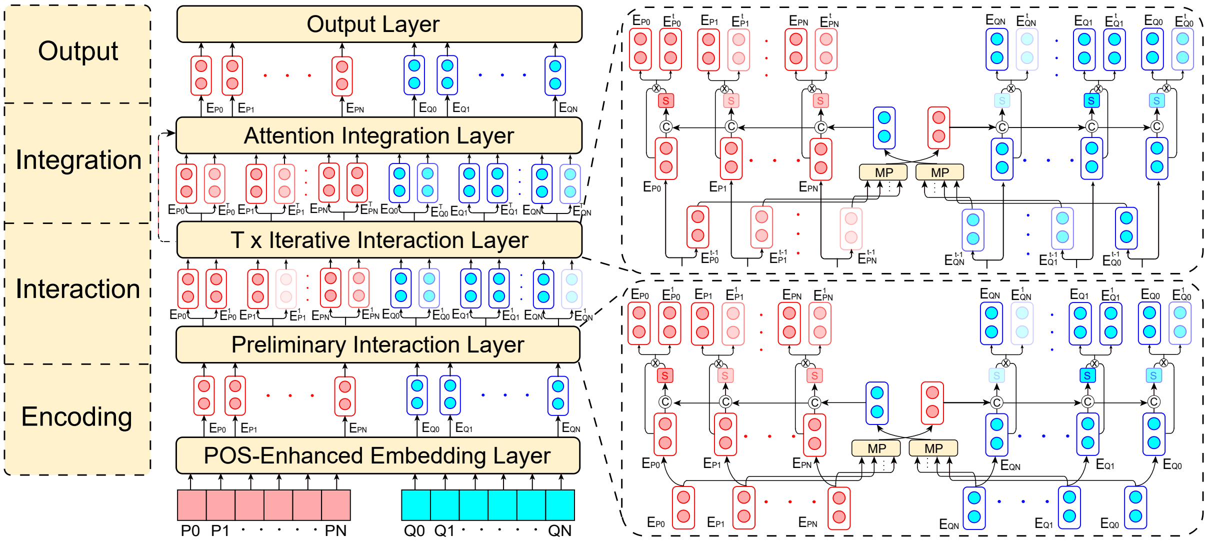

Figure 1: Overview of *POI-Net*. $s, c, \times, MP$ donate the normalized attention score, similarity calculation, scalar multiplication, and max pooling operation respectively. The shade of color represent the contribution of corresponding embedding to operating question.

categories in discriminative MRC. Others utilize excessively detailed design for one special MRC subcategory at decoding aspect (Sun et al., 2019b; Zhang et al., 2020a), lacking the universality for overall discriminative MRC.

To fill in above gap in unified modeling for different discriminative MRCs, based on core focuses of extractive and multi-choice MRC, we design two complementary reading strategies at both encoding and decoding aspects. The encoding design enhances token *linguistic* representation at local level, which is especially effective for extractive MRC. The explicit possession of word part-of-speech (POS) attribute of human leads to precise answer extraction. In the extractive sample from Table 1, human extracts golden answer span precisely because "*London Exhibition*" is a proper noun (NNP) corresponding to interrogative qualifier (WDT) "*Where*" in the question, while imprecise words like "*1862*" (cardinal number, CD) and "*exhibited*" (past tense verb, VBD) predicted by machines will not be retained. Thus, we inject word POS attribute explicitly in embedding form.

The decoding design simulates human *reconsideration* and *integration* abilities at global level, with especial effect for multi-choice MRC. To handle compound questions with limited attention, human will highlight critical information in turns, and update recognition and attention distribution iteratively. Inspired by above *reconsideration* strategy, we design *Iterative Co-Attention Mechanism* with no additional parameter, which iteratively executes the interaction between passage and question-option ($Q-O$) pair globally in turns. In the multi-choice example from Table 1, during the first interaction, model may only focus on texts related to rough impression of $Q-O$ pair such as "*Green Scenes*", ignoring plentiful but scattered critical information. But with sufficient iterative interaction, model can ultimately collect all detailed evidence (bold in Table 1). Furthermore, we explore a series of attention *integration* strategies for captured evidence among interaction turns.

We combine two above methods and propose a novel model called *POI-Net* (**POS**-Enhanced **I**terative Co-Attention **Net**work), to alleviate the gap between machines and humans on discriminative MRC. We evaluate our model on two multi-choice MRC benchmarks, RACE (Lai et al., 2017) and DREAM (Sun et al., 2019a); and two extractive MRC benchmarks, SQuAD 1.1 (Rajpurkar et al., 2016) and SQuAD 2.0 (Rajpurkar et al., 2018), obtaining consistent and significant improvements, with nearly zero additional parameters.

2 Our Model

We aim to design a lightweight, universal and effective model architecture for various subcategories of discriminative MRC, and the overview of our model is shown in Figure 1, which consists of four main processes: Encoding (§2.1), Interaction (§2.2), Integration (§2.3) and Output (§2.4).

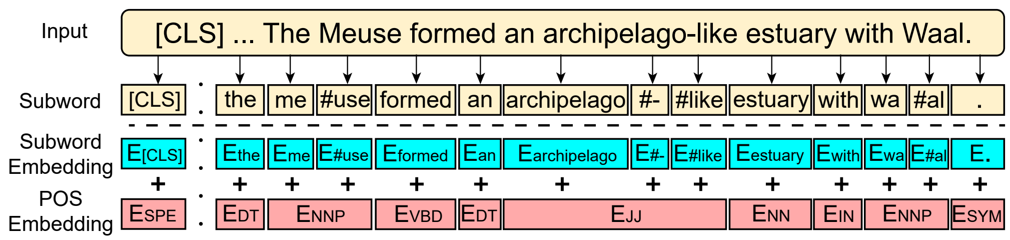

Figure 2: The input representation flow of *POI-Net*. The subscripts of *POS Embedding* are POS tags of input words.

2.1 POS-Enhanced Encoder

Based on pre-trained contextualized encoder ALBERT (Lan et al., 2020), we encode input tokens with an additional POS embedding layer, as Figure 2 shows. Since the input sequence will be tokenized into subwords in the contextualized encoder, we tokenize sequences in word-level with *nltk* tokenizer (Bird et al., 2009) additionally and implement *POS-Enhanced Encoder*, where each subword in a complete word will share the same POS tag.

In detail, input sequences are fed into *nltk* POS tagger to obtain the POS tag of each word such as "JJ". Subject to Penn Treebank style, our adopted POS tagger has 36 POS tag types. Considering on the specific scenarios in discriminative MRC, we add additional SPE tag for special tokens (i.e., $[CLS]$, $[SEP]$), PAD tag for padding tokens and ERR tag for potential unrecognized tokens. Appendix A shows detailed description of POS tags.

The input embedding in our model is the normalized sum of *Subword Embedding* and *POS Embedding*. Following the basic design in embedding layers of BERT-style models, we retain Token Embedding E_t, Segmentation Embedding E_s and Position Embedding E_p in subword-level, constituting *Subword Embedding*. For *POS Embedding* E_{POS}, we implement another embedding layer with the same embedding size to *Subword Embedding*, guaranteeing all above indicator embeddings are in the same vector space. Formulaically, the input embedding E can be represented as:

$$E = Norm(E_t + E_s + E_p + E_{POS}),$$

where $Norm()$ is a layer normalization function (Ba et al., 2016).

2.2 Iterative Co-Attention Mechanism

POI-Net employs a lightweight *Iterative Co-Attention* module to simulate human inner reconsidering process, with **no** additional parameter.

2.2.1 Preliminary Interaction

POI-Net splits all N input token embeddings into passage domain (P) and question (or $Q-O$ pair) domain (Q) to start $P-Q$ interactive process. To generate the overall impression of the given passage or question like humans, *POI-Net* concentrates all embeddings in corresponding domain into one *Concentrated Embedding* by max pooling:

$$CE_P^1 = MaxPooling(E_{P0}, ..., E_{PN}) \in \mathbb{R}^H,$$

$$CE_Q^1 = MaxPooling(E_{Q0}, ..., E_{QN}) \in \mathbb{R}^H,$$

where H is the hidden size, PN/QN is the token amount of P/Q domain. Then *POI-Net* calculates the similarity between each token in E_P/E_Q and CE_Q^1/CE_P^1, to generate attention score s for each token contributing to the $P-Q$ pair. In detail, we use cosine similarly for calculation:

$$s_{P0}^1, ..., s_{PN}^1 = Cosine([E_{P0}, ..., E_{PN}], CE_Q^1),$$

$$s_{Q0}^1, ..., s_{QN}^1 = Cosine([E_{Q0}, ..., E_{QN}], CE_P^1).$$

We normalize these scores to $[0, 1]$ by min-max scaling, then execute dot product with corresponding input embeddings:

$$E_{Pi}^1 = \hat{s}_{Pi}^1 \cdot E_{Pi}, \quad E_{Qi}^1 = \hat{s}_{Qi}^1 \cdot E_{Qi},$$

where $\hat{s}_{Pi}$ is the normalized attention score of i-th passage token embedding, E_{Pi}^1 is the attention-enhanced embedding of i-th passage token after preliminary interaction (the 1-st turn interaction).

2.2.2 t-th Turn Interaction

To model human reconsideration ability between passage and question in turns, we add iterable modules with co-attention mechanism, as the *Iterative Interaction Layer* in Figure 1. Detailed processes in the t-th turn interaction are similar to preliminary interaction:

$$CE_Q^t = MaxPooling(E_{Q0}^{t-1}, ..., E_{QN}^{t-1}) \in \mathbb{R}^H,$$

$$CE_P^t = MaxPooling(E_{P0}^{t-1}, ..., E_{PN}^{t-1}) \in \mathbb{R}^H,$$

$$s^t_{P0}, ..., s^t_{PN} = Cosine([E_{P0}, ..., E_{PN}], CE^t_Q),$$
$$s^t_{Q0}, ..., s^t_{QN} = Cosine([E_{Q0}, ..., E_{QN}], CE^t_P),$$
$$E^t_{Pi} = \hat{s}^t_{Pi} \cdot E_{Pi}, \quad E^t_{Qi} = \hat{s}^t_{Qi} \cdot E_{Qi}.$$

Note that, during all iteration turns, we calculate attention scores with the original input embedding E instead of attention-enhanced embedding E^{t-1} from the (t-1)-th turn, due to:

1) There is no further significant performance improvement by replacing E with E^{t-1} ($< 0.2\%$ on base size model), comparing to adopted method;

2) With the same embedding E, attention integration in §2.3 can be optimized into attention score integration, which is computationally efficient with no additional embedding storage[1].

2.3 Attention Integration

Human recommends to integrate all critical information from multiple turns for a comprehensive conclusion, instead of discarding all findings from previous consideration. In line with above thought, *POI-Net* returns attention-enhanced embedding $E^t = \hat{s}^t \cdot E$ of each turn (we only store $\hat{s}^t$ in an optimized method), and integrates them with specific strategies. We design four integration strategies according to the contribution proportion of each turn and adopt *Forgetting Strategy* ultimately.

- **Average Strategy**: The attention network treats normalized attention score $\hat{s}^t$ of each turn equally, and produces the ultimate representation vector $\mathbf{R}$ with average value of $\hat{s}^t$:
$$\mathbf{R} = \frac{1}{T}\sum_{t=1}^{T} \hat{s}^t \cdot E \ \in \mathbb{R}^{N\times H},$$
where T is the total amount of iteration turns.

- **Weighted Strategy**: The attention network treats $\hat{s}^t$ with two normalized weighted coefficients β^t_P, β^t_Q, which measure the contribution of the t-th turn calculation:
$$\mathbf{R} = \frac{\sum_{t=1}^T \beta^t_P \hat{s}^t_P}{\sum_{t=1}^T \beta^t_P} \cdot E_P + \frac{\sum_{t=1}^T \beta^t_Q \hat{s}^t_Q}{\sum_{t=1}^T \beta^t_Q} \cdot E_Q,$$
$$\tilde{\beta}^t_P = Max(s^{t-1}_{Q0}, ..., s^{t-1}_{QN}),$$
$$\tilde{\beta}^t_Q = Max(s^{t-1}_{P0}, ..., s^{t-1}_{PN}),$$
$$\beta^t_P = \frac{\tilde{\beta}^t_P + 1}{2}, \ \beta^t_Q = \frac{\tilde{\beta}^t_Q + 1}{2},$$
where $s^0_{Pi} = s^0_{Qi} = 1.0$. The design motivation for β^t_P, β^t_Q is intuitive: when *Concentrated Embedding* CE^t_Q/CE^t_P (calculating attention score at the t-th turn) has higher confidence (behaving as higher maximum value in s^{t-1}_Q/s^{t-1}_P due to max pooling calculation), system should pay more attention to input embedding E^t_P/E^t_Q at the t-th turn[2].

- **Forgetting Strategy**: Since human will partly forget knowledge from previous consideration and focus on findings at current turn, we execute normalization operation of attention scores from two most previous turns iteratively:
$$\mathbf{R} = \frac{\mathbf{s^T_P} + \beta^t_P \hat{s}^T_P}{1 + \beta^T_P} \cdot E_P + \frac{\mathbf{s^T_Q} + \beta^t_Q \hat{s}^T_Q}{1 + \beta^T_Q} \cdot E_Q,$$
$$\mathbf{s^T_P} = \frac{\mathbf{s^{T-1}_P} + \beta^t_P \hat{s}^{T-1}_P}{1 + \beta^{T-1}_P},$$
$$\mathbf{s^T_Q} = \frac{\mathbf{s^{T-1}_Q} + \beta^t_Q \hat{s}^{T-1}_Q}{1 + \beta^{T-1}_Q}.$$
During the iterative normalization, the ultimate proportion of attention scores from previous turns will be diluted gradually, which simulates the effect of forgetting strategy[3].

- **Intuition Strategy**: In some cases, human can solve simple questions in intuition without excessive consideration, thus we introduce two attenuation coefficients α^t_P, α^t_Q for attention scores from the t-th turn, which decrease gradually as the turn of iteration increases:
$$\mathbf{R} = \frac{\sum_{t=1}^T \alpha^t_P \hat{s}^t_P}{\sum_{t=1}^T \alpha^t_P} \cdot E_P + \frac{\sum_{t=1}^T \alpha^t_Q \hat{s}^t_Q}{\sum_{t=1}^T \alpha^t_Q} \cdot E_Q,$$
$$\alpha^t_P = \prod_{i=1}^{t} \beta^i_P, \ \alpha^t_Q = \prod_{i=1}^{t} \beta^i_Q.$$

[1]Approximate 15.3% training time is saved on average for each iteration turn.

[2]Setting β^t_P/β^t_Q as learnable parameters cannot bring further improvement since the contribution proportion of each turn varies with the specific circumstance of input samples.

[3]Method of activation functions in LSTM (Hochreiter and Schmidhuber, 1997) may filter out information completely in one single-turn calculation, which cannot bring consistent improvement in our experiments.

2.4 Adaptation for Discriminative MRC

2.4.1 Multi-choice MRC

The input sequence for multi-choice MRC is $[CLS]\ P\ [SEP]\ Q + O_i\ [SEP]$, where $+$ denotes concatenation, O_i denotes the i-th answer options. In *Output Layer*, the representation vector $\mathbf{R} \in \mathbb{R}^{N \times H}$ is fed into a max pooling operation to generate general representation:

$$R = MaxPooling(\mathbf{R}) \in \mathbb{R}^H.$$

Then a linear softmax layer is employed to calculate probabilities of options, and standard Cross Entropy Loss is employed as the total loss. Option with the largest probability is determined as the predicted answer.

2.4.2 Extractive MRC

The input sequence for extractive MRC can be represented as $[CLS]\ P\ [SEP]\ Q\ [SEP]$, and we use a linear softmax layer to calculate start and end token probabilities in *Output Layer*. The training object is the sum of Cross Entropy Losses for the start and end token probabilities:

$$\mathcal{L} = y_s \cdot log(s) + y_e \cdot log(e),$$

$$s, e = softmax(Linear(\mathbf{R})) \in \mathbb{R}^N,$$

where s/e are the start/end probabilities for all tokens and y_s/y_e are the start/end targets.

For answer prediction, since some benchmarks have unanswerable questions, we first score the span from the i-th token to the j-th token as:

$$score_{ij} = s_i + e_j, \quad 0 \leq i \leq j \leq N,$$

then the span with the maximum score $score_{has}$ is the predicted answer. The score of null answer is: $score_{no} = s_0 + e_0$, where the 0-th token is $[CLS]$. The final score is calculated as $score_{has} - score_{no}$, and a threshold δ is set to determine whether the question is answerable, which is heuristically computed in linear time. *POI-Net* predicts the span with the maximum score if the final score is above the threshold, and null answer otherwise.

3 Experiments

3.1 Setup & Dataset

The experiments are run on 8 NVIDIA Tesla P40 GPUs and the implementation of *POI-Net* is based on the Pytorch implementation of ALBERT (Paszke et al., 2019). We set the maximum iteration turns in *Iterative Co-Attention* as 3. Table 2 shows the hyper-parameters of *POI-Net* achieving reported results. As a supplement, the warmup rate is 0.1 for all tasks.

Hyperparam	LR	MSL	BS	TE	SS
DREAM	1e-5	512	24	4	400
RACE	1e-5	512	32	2	4000
SQuAD 1.1	1e-5	512	24	2	2000
SQuAD 2.0	1e-5	512	24	2	4000

Table 2: The fine-tuning hyper-parameters of *POI-Net*. LR: learning rate, MSL: maximum sequence length, BS: batch size, TE: training epochs, SS: save steps.

We evaluate *POI-Net* on two multi-choice MRC benchmarks: RACE (Lai et al., 2017), DREAM (Sun et al., 2019a), and two extractive MRC benchmarks: SQuAD 1.1 (Rajpurkar et al., 2016) and SQuAD 2.0 (Rajpurkar et al., 2018). The detailed introduction is shown as following:

RACE is a large-scale multi-choice MRC task collected from English examinations which contains nearly 100K questions. The passages are in the form of articles and most questions need contextual reasoning, and the domains of passages are diversified.

DREAM is a dialogue-based dataset for multi-choice MRC, containing more than 10K questions. The challenge of the dataset is that more than 80% of the questions are non-extractive and require reasoning from multi-turn dialogues.

SQuAD 1.1 is a widely used large-scale extractive MRC benchmark with more than 107K passage-question pairs, which are produced from Wikipedia. Models are asked to extract precise word span from the Wikipedia passage as the answer of the given passage.

SQuAD 2.0 retains the questions in SQuAD 1.1 with over 53K unanswerable questions, which are similar to answerable ones. For SQuAD 2.0, models must not only answer questions when possible, but also abstain from answering when the question is unanswerable with the paragraph.

3.2 Results

We take accuracy as evaluation criteria for multi-choice benchmarks, while exact match (EM) and

[4]Due to the test sets of SQuAD 1.1 and SQuAD 2.0 are not open for free evaluation with different random seeds, we report the results on development set instead.

Model	DREAM		RACE		SQuAD 1.1		SQuAD 2.0	
	Dev	Test	Dev(M/H)	Test(M/H)	EM	F1	EM	F1
BERT$_{base}$ (Devlin et al., 2019)	63.4	63.2	64.6 (– / –)	65.0 (71.1 / 62.3)	80.8	88.5	77.6	80.4
ALBERT$_{base}$ (Lan et al., 2020)	64.5	64.4	64.0 (– / –)	– (– / –)	82.3	89.3	77.1	80.0
BERT$_{large}$ (Devlin et al., 2019)	66.0	66.8	72.7 (76.7 / 71.0)	72.0 (76.6 / 70.1)	85.5	92.2	82.2	85.0
SG-Net (Zhang et al., 2020c)	–	–	– (– / –)	74.2 (78.8 / 72.2)	–	–	85.6	88.3
RoBERTa$_{large}$ (Liu et al., 2019)	85.4	85.0	– (– / –)	83.2 (86.5 / 81.8)	–	–	86.5	89.4
RoBERTa$_{large}$+MMM (Jin et al., 2020)	88.0	88.9	– (– / –)	85.0 (89.1 / 83.3)	–	–	–	–
ALBERT$_{xxlarge}$ (Lan et al., 2020)	89.2	88.5	– (– / –)	86.5 (89.0 / 85.5)	88.3	94.1	85.1	88.1
ALBERT$_{xxlarge}$ + DUMA (Zhu et al., 2020)	89.9	90.4	88.1 (– / –)	88.0 (90.9 / 86.7)	–	–	–	–
ALBERT$_{base}$ (rerun)	65.7	65.6	67.9 (72.3 / 65.7)	67.2 (72.1 / 65.2)	82.7	89.9	77.9	81.0
POI-Net on ALBERT$_{base}$	68.6	68.5	72.4 (76.3 / 70.0)	71.0 (75.7 / 69.0)	84.5	91.3	79.5	82.7
ALBERT$_{xxlarge}$ (rerun)	88.7	88.3	86.6 (89.4 / 85.2)	86.5 (89.2 / 85.4)	88.2	93.9	85.4	88.5
POI-Net on ALBERT$_{xxlarge}$	**90.0**	90.3	**88.1 (91.2 / 86.3)**	**88.3 (91.5 / 86.8)**	**89.5**	**95.0**	**87.7**	**90.6**

Table 3: Results of BERT-style models on DREAM, RACE, SQuAD 1.1 and SQuAD 2.0. Results in the first domain are from the leaderboards and corresponding papers[4].

a softer metric F1 score for extractive benchmarks. The average results of three random seeds are shown in Table 3, where we only display several BERT-style models with comparable parameters. Appendix B reports the complete comparison results with other public works on each benchmark.

The results show that, for multi-choice benchmarks, our model outperforms most baselines and comparison works, and passes the significance test (Zhang et al., 2021) with $p - value < 0.01$ in DREAM (2.0% average improvement) and RACE (1.7% average improvement). And for extractive benchmarks, though the performance of baseline ALBERT is strong, our model still boosts it essentially (1.3% average improvement on EM for SQuAD 1.1 and 2.3% for SQuAD 2.0). Furthermore, we report the parameter scale and training/inference time costs in §4.4.

4 Ablation Studies

In this section, we implement *POI-Net* on ALBERT$_{base}$ for further discussions, and such settings have the similar quantitative tendency to *POI-Net* on ALBERT$_{xxlarge}$.

4.1 Ablation

Model	RACE	SQuAD 1.1	
	Acc	EM	F1
Baseline (ALBERT$_{base}$)	67.88	82.66	89.91
POI-Net on ALBERT$_{base}$	72.44	84.48	91.28
- POS Embedding	71.74	83.51	90.64
- Iterative Co-Attention	69.02	83.65	90.77
Baseline (rerun BERT$_{base}$)	64.73	81.21	88.84
POI-Net on BERT$_{base}$	68.02	83.43	90.47

Table 4: Ablation studies on RACE and SQuAD 1.1.

To evaluate the contribution of each component in *POI-Net*, we perform ablation studies on RACE and SQuAD 1.1 development sets and report the average results of three random seeds in Table 4. The results indicate that, both *POS Embedding* and *Iterative Co-Attention Mechanism* provide considerable contributions to *POI-Net*, but in different roles for certain MRC subcategory.

For multi-choice MRC like RACE, *Iterative Co-Attention Mechanism* contributes much more than *POS Embedding* (3.86% v.s. 1.14%), since multi-choice MRC requires to highlight and integrate critical information in passages comprehensively. Therefore, potential omission of critical evidence may be fatal for answer prediction, which is guaranteed by *Iterative Co-Attention Mechanism*, while precise evidence span boundary and POS attributes are not as important as the former.

On the contrary, simple *POS Embedding* even brings a little more improvement than the well-designed *Iterative Co-Attention* (0.99% v.s. 0.85% on EM) for extractive MRC. In these tasks, model focuses on answer span extraction with precise boundaries, and requires to discard interference words which not exactly match questions, such as redundant verbs, prepositions and infinitives ("*politically and socially unstable*" instead of "***to be** politically and socially unstable*"), or partial interception of proper nouns ("*Seljuk Turks*" instead of "*Turks*"). With the POS attribute of each word, *POI-Net* locates the boundaries of answer spans precisely[5]. Since extractive MRC does not require comprehensive information integration like multi-

[5]Note that, the improvement of *POI-Net* on EM score is consistently higher than F1 score, as corroboration.

choice MRC, the improvement from *Iterative Co-Attention Mechanism* is less significant.

Besides, we also implement *POI-Net* on other contextualized encoders like BERT, and achieve significant improvements as Table 4 shows. The consistent and significant improvements over various baselines verify the universal effectiveness of *POI-Net*.

4.2 Role of POS Embedding

POS Type	Golden Answer	POI-Net	Baseline
NN	11192	11254	11504
CD	3511	3723	3816
NNS	2875	2812	2743
JJ	1654	1671	1774
IN	396	308	242
VBN	348	321	299
RB	339	315	284
VBG	331	328	293

Table 5: The POS type statistics of boundary words in golden answer, predicted answer by *POI-Net* and baseline ALBERT$_{base}$. We only display POS types whose occurrence is higher than 300.

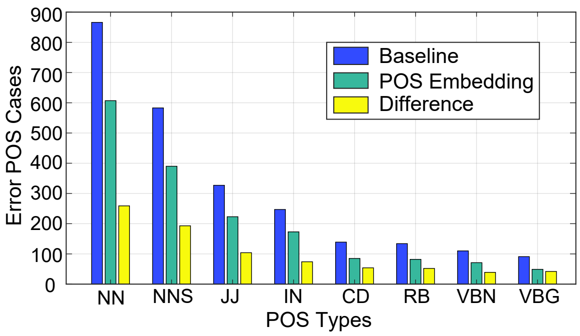

Figure 3: Error POS classification case statistics of *POI-Net* and baseline. For explanation, the first square pillar (Height: 866) means, there are 866 cases whose POS type of boundary word in golden answer is "NN", but the baseline predicts an error word in a non-"NN" type.

To study how *POS Embedding* enhances token representation, we make a series of statistics on SQuAD 1.1 development set about: 1) POS type of boundary words from predicted spans, as Table 5 shows; 2) error POS classification of *POI-Net* and its baseline ALBERT$_{base}$, as Figure 3 shows.

The statistical results show, with *POS Embedding*, the overall distribution of the POS types of answer boundary words predicted by *POI-Net* is more similar to golden answer, compared with its baseline; and the amount of error POS classification cases by *POI-Net* also reduces significantly. And there are also two further findings:

1) The correction proportion of error POS classification (8.09%) is much higher than correction proportion of overall error predictions (1.82%) in *POI-Net*, which indicates the correction of POS classification benefits mostly from the perception of word POS attributes by *POS Embedding*, instead of the improvement on overall accuracy.

2) Though answers in SQuAD 1.1 incline to distribute in several specific POS types ("NN", "CD", "NNS" and "JJ"), *POS Embedding* prompts model to consider words in each POS type more equally than the baseline, and the predicted proportions of words in rarer POS type ("IN", "VBN", "RB", "VBG" and so on) increase.

4.3 Research on the Robustness of POS Embedding

Robustness is one of the important indicators to measure model performance, when there is numerous rough data or resource in applied tasks. To measure the anti-interference of *POS Embedding*, we randomly modify part of POS tags from *nltk* POS tagger to error tags, and the results on SQuAD 1.1 development set are shown in Table 6.

Model	EM	F1
Baseline (ALBERT$_{base}$)	82.66	89.91
POI-Net on ALBERT$_{base}$	84.48	91.28
5% error POS tags	84.35	91.21
10% error POS tags	84.06	91.05
20% error POS tags	83.87	90.80
- POS Embedding	83.51	90.64

Table 6: Results of robustness research of POS Embedding on dev sets from SQuAD 1.1.

The results indicate that, *POI-Net* possesses satisfactory *POS Embedding* robustness, and the improvement brought by *POS Embedding* will not suffer a lot with a slight disturbance (5%). We argue that the robustness of *POI-Net* may benefit from the integration with other contextualized embeddings, such as Token Embedding E_t which encodes the contextual meaning of current word or subword. Though more violent interference (20%) may further hurt token representations, existing mature POS taggers achieve 97% + accuracy, which can prevent the occurrence of above situations.

4.4 Role of Iterative Co-Attention Mechanism

To explore the most suitable integration strategy and maximum iteration turn in *Iterative Co-Attention Mechanism*, we implement our proposed strategies with different maximum iteration turns, together with a baseline replacing *Iterative Co-Attention* mechanism by a widely used Multi-head Co-Attention mechanism (Devlin et al., 2019; Zhang et al., 2020a, 2021) for comparison in Figure 4. We take RACE as the evaluated benchmark due to the significant effect of attention mechanism to multi-choice MRC.

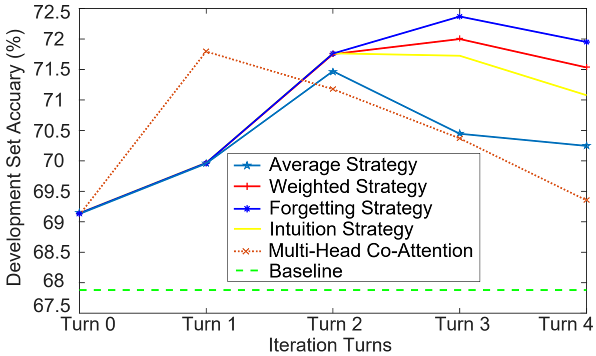

Figure 4: Comparative experiments on *Iterative Co-Attention Mechanism*. When iteration turn is 0, the model is equivalent to baseline with *POS Embedding*.

As the figure shows, forgetting strategy leads to the best performance, with slight improvement than weighted strategy. Both these two strategies are in line with the logical evidence integration in human reconsidering process, while average strategy and intuition strategy may work against common human logic. From the trends of four strategies in multiple iterations, we conclude that 2 or 3 iteration turns for *Iterative Co-Attention* lead to an appropriate result, due to:

1) Fewer iteration turns may lead to inadequate interaction between passage and question, and model may focus on rough cognition instead of exhaustive critical information;

2) Excessive iteration turns may lead to over-integration of information, declining the contribution by real critical evidence.

Compared to the typical Multi-head Co-Attention mechanism, our proposed *Iterative Co-Attention* mechanism obtains higher performance with more iterations, indicating it has stronger iterative reconsideration ability.

Besides, *Iterative Co-Attention* defeats Multi-head Co-Attention on both parameter size and training time cost. As the parameter comparison in Table 7 shows, *POI-Net* basically brings no additional parameter except an linear embedding layer for *POS Embedding*. Multi-head Co-Attention mechanism and models based on it (like DUMA in Table 3) introduces much more parameters, with slightly

Model	**Parameters**
ALBERT$_{base}$ (Lan et al., 2020)	12M
ALBERT$_{base}$ (rerun)	11.14M
Multi-head Co-Attention on ALBERT$_{base}$	17.94M
POI-Net on ALBERT$_{base}$	**11.15M**
ALBERT$_{xxlarge}$ (Lan et al., 2020)	235M
ALBERT$_{xxlarge}$ (rerun)	212.29M
Multi-head Co-Attention on ALBERT$_{xxlarge}$	404.50M
POI-Net on ALBERT$_{xxlarge}$	**212.30M**

Table 7: Training parameters in *POI-Net* and baselines.

lower performance. We also record time costs on RACE for one training epoch on ALBERT$_{base}$, *Iterative Co-Attention* costs $54, 62, 72, 83, 96$ minutes from 0-turn iteration to 4-turn iterations, while Multi-head Co-Attention costs $54, 65, 76, 89, 109$ minutes instead, with 8.3% increase on average.

4.5 Visualization

We perform a visualization display for discriminative MRC examples in Table 1, as Figure 5 shows. For the extractive example, benefited from *POS Embedding*, *POI-Net* predicts the precise answer span, based on the interrogative qualifier "*where*" and POS attributes of controversial boundary tokens *"exhibited", "at", "London", "Exhibition", "1862"*.

And for the multi-choice example, without proposed *Iterative Co-Attention Mechanism*, the overall distribution of attention is more scattered. The baseline can only notice special tokens like $[CLS]$ at the 0-th turn, and even interrogative qualifier "*how*" due to the similar usage to "*what*" in the question. With the execution of *Iterative Co-Attention*, *POI-Net* pays more attention on discrete critical words like "*Green Scenes*" and "*events*" at the 1-st turn, "*series*" and "*focusing*" at the 2-nd turn and "*greener lifestyle*" at the 3-rd turn. After the integration of all above critical evidence, *POI-Net* predicts the golden option ultimately.

5 Related Studies

5.1 Semantic and Linguistic Embedding

To cope with challenging MRC tasks, numerous powerful pre-trained language models (PLMs) have been proposed (Devlin et al., 2019; Lewis et al., 2020; Raffel et al., 2020). Though advanced PLMs demonstrate strong ability in contextual representation, the lack of explicit *semantic* and *linguistic* clues leads to the bottleneck of previous works.

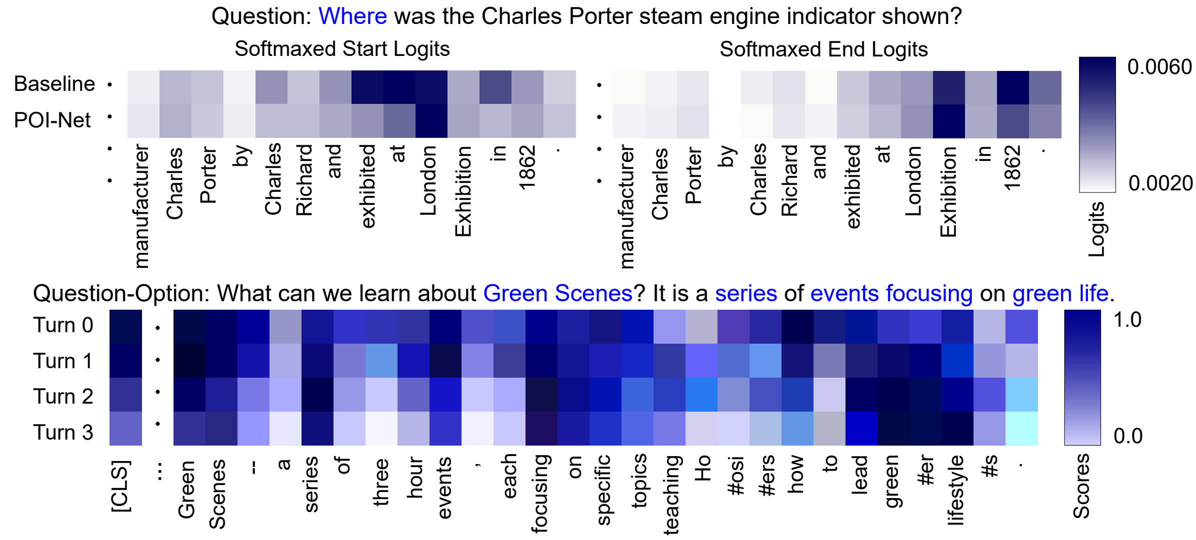

Figure 5: Visualization of *POI-Net* and its baseline on extractive example (upper) and multi-choice example (lower) in Table 1. The indicator for extractive example is softmaxed logit, and for multi-choice example is normalized attention score $\hat{s}_P^t$.

Benefited from the development of semantic role labeling (Li et al., 2018) and dependency syntactic parsing (Zhou and Zhao, 2019), some researchers focus on enhancing semantic representations. Zhang et al. (2020b) strengthen token representation by fusing semantic role labels, while Zhang et al. (2020c) and Bai et al. (2021) implement additional self attention layers to encode syntactic dependency. Furthermore, Mihaylov and Frank (2019) employ multiple discourse-aware semantic annotations for MRC on narrative texts.

Instead of semantic information, we pay attention to more accessible part-of-speech (POS) information, which has been widely used into non-MRC fields, such as open domain QA (Chen et al., 2017), with much lower pre-processing calculation consumption but higher accuracy (Bohnet et al., 2018; Strubell et al., 2018; Zhou et al., 2020). However, previous application of POS attributes mostly stays in primitive and rough embedding methods (Huang et al., 2018), leading to much slighter improvement than proposed *POI-Net*.

5.2 Attention Mechanism

In discriminative MRC field, various attention mechanisms (Raffel and Ellis, 2015; Seo et al., 2017; Wang et al., 2017; Vaswani et al., 2017) play increasingly important roles. Initially, attention mechanism is mainly adopted on extractive MRC (Yu et al., 2018; Cui et al., 2021), such as multiple polishing of answer spans (Xiong et al., 2017) and multi-granularity representations generation (Zheng et al., 2020; Chen et al., 2020). Recently, researchers notice its special effect for multi-choice MRC. Zhang et al. (2020a) model domains bidirectionally with dual co-matching network, Jin et al. (2020) use multi-step attention as classifier, and Zhu et al. (2020) design multi-head co-attentions for collaborative interactions.

We thus propose a universal *Iterative Co-Attention* mechanism, which performs interaction between paired input domains iteratively, to hopefully enhance discriminative MRC. Unlike other works introducing numerous parameters by complicated attention network (Zhang et al., 2020a), our *POI-Net* is more effective and efficient with almost no introduction of additional parameters.

6 Conclusion

In this work, we propose **POS**-Enhanced **I**terative Co-Attention **Net**work (*POI-Net*), as a lightweight unified modeling for multiple subcategories of discriminative MRC. *POI-Net* utilizes *POS Embedding* to encode POS attributes for the preciseness of answer boundary, and *Iterative Co-Attention Mechanism* with integration strategy is employed to highlight and integrate critical information at decoding aspect, with almost no additional parameter. As the first effective and unified modeling with pertinence for different types of discriminative MRC, evaluation results on four extractive and multi-choice MRC benchmarks consistently indicate the general effectiveness and applicability of our model.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Seohyun Back, Sai Chetan Chinthakindi, Akhil Kedia, Haejun Lee, and Jaegul Choo. 2020. Neurquri: Neural question requirement inspector for answerability prediction. In *ICLR*.

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-BERT: Improving pre-trained transformers with syntax trees. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3011–3020, Online. Association for Computational Linguistics.

Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani. 2020. A survey on machine reading comprehension systems. *arXiv preprint arXiv:2001.01582*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Bernd Bohnet, Ryan McDonald, Gonçalo Simões, Daniel Andor, Emily Pitler, and Joshua Maynez. 2018. Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2642–2652, Melbourne, Australia. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Nuo Chen, Fenglin Liu, Chenyu You, Peilin Zhou, and Yuexian Zou. 2020. Adaptive bi-directional attention: Exploring multi-granularity representations for machine reading comprehension.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yiming Cui, Wei-Nan Zhang, Wanxiang Che, Ting Liu, and Zhigang Chen. 2021. Understanding attention in machine reading comprehension.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, and Ming Zhou. 2018. Reinforced mnemonic reader for machine reading comprehension. In *IJCAI*.

Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. 2018. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. In *International Conference on Learning Representations*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Yufan Jiang, Shuangzhi Wu, Jing Gong, Yahui Cheng, Peng Meng, Weiliang Lin, Zhibo Chen, and Mu li. 2020. Improving machine reading comprehension with single-choice decision and transfer learning.

Di Jin, Shuyang Gao, Jiun-Yu Kao, Tagyoung Chung, and Dilek Hakkani-tur. 2020. Mmm: Multi-stage multi-task learning for multi-choice reading comprehension. In *AAAI*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *TACL*, 6:317–328.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Zuchao Li, Shexia He, Jiaxun Cai, Zhuosheng Zhang, Hai Zhao, Gongshen Liu, Linlin Li, and Luo Si. 2018. A unified syntax-aware framework for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2401–2411, Brussels, Belgium. Association for Computational Linguistics.

Rui Liu, Junjie Hu, Wei Wei, Zi Yang, and Eric Nyberg. 2017. Structural embedding of syntactic trees for machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 815–824, Copenhagen, Denmark. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Todor Mihaylov and Anette Frank. 2019. Discourse-aware semantic self-attention for narrative reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2541–2552, Hong Kong, China. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Colin Raffel and Daniel P. W. Ellis. 2015. Feed-forward networks with attention can solve some long-term memory problems.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038, Brussels, Belgium. Association for Computational Linguistics.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019a. DREAM: A challenge data set and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2019b. Improving machine reading comprehension with general reading strategies. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2633–2643, Minneapolis, Minnesota. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, volume 30.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198, Vancouver, Canada. Association for Computational Linguistics.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. In *ICLR*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *International Conference on Learning Representations*.

Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2020a. DCMN+: Dual co-matching network for multi-choice reading comprehension. In *AAAI*.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020b. Semantics-aware BERT for language understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9628–9635. AAAI Press.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020c. Sg-net: Syntax-guided machine reading comprehension. *AAAI*.

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2021. Retrospective reader for machine reading comprehension. In *AAAI*.

Bo Zheng, Haoyang Wen, Yaobo Liang, Nan Duan, Wanxiang Che, Daxin Jiang, Ming Zhou, and Ting Liu. 2020. Document modeling with graph attention networks for multi-grained machine reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6708–6718, Online. Association for Computational Linguistics.

Junru Zhou, Zhuosheng Zhang, Hai Zhao, and Shuailiang Zhang. 2020. LIMIT-BERT : Linguistics informed multi-task BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4450–4461, Online. Association for Computational Linguistics.

Junru Zhou and Hai Zhao. 2019. Head-Driven Phrase Structure Grammar parsing on Penn Treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408, Florence, Italy. Association for Computational Linguistics.

Pengfei Zhu, Hai Zhao, and Xiaoguang Li. 2020. Dual multi-head co-attention for multi-choice reading comprehension. *arXiv preprint arXiv:2001.09415*.

A Part-Of-Speech Tags List

In this appendix, we list all 39 POS tags (including POS tags from *nltk* POS tagger and defined by us) in Table 9.

B Complete Comparison Results on Benchmarks

We show complete public works on DREAM, RACE, SQuAD 1.1 and SQuAD in this appendix, as Tables 8 10, 11 and 12 show.

The results show that, our *POI-Net* outperforms most of comparison models and baselines, expect models: 1) with massive and incomparable parameters like T5 (Raffel et al., 2020) and Megatron-BERT (Shoeybi et al., 2019); 2) in more advanced baseline architecture like XLNet (Yang et al., 2019), ELECTRA (Clark et al., 2020); 3) in special model design for one single subcategory of discriminative MRC task (Zhang et al., 2021).

Model	Dev	Test
FTLM++ (Sun et al., 2019a)	58.1	58.2
BERT$_{base}$ (Devlin et al., 2019)	63.4	63.2
BERT$_{large}$ (Devlin et al., 2019)	66.0	66.8
XLNet$_{large}$ (Yang et al., 2019)	–	72.0
RoBERTa$_{large}$ (Liu et al., 2019)	85.4	85.0
RoBERTa$_{large}$ + MMM (Jin et al., 2020)	88.0	88.9
ALBERT$_{xxlarge}$ + DUMA (Zhu et al., 2020)	89.9	90.4
ALBERT$_{xxlarge}$ + DUMA + MTL	–	91.8
ALBERT$_{base}$ (rerun)	65.7	65.6
POI-Net on ALBERT$_{base}$	68.6	68.5
ALBERT$_{xxlarge}$ (rerun)	89.2	88.5
POI-Net on ALBERT$_{xxlarge}$	90.0	90.3

Table 8: Public submissions on DREAM. The results in the first domain are from the leaderboard. MTL denotes multi-task learning.

POS Tag	Meaning
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	To
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb
SPE	Special tokens: [CLS], [SEP]
PAD	Padding tokens
ERR	Unrecognized tokens

Table 9: The complete list for all POS tags in *POI-Net*.

Model	**Dev (M / H)**	**Test (M / H)**
BERT$_{base}$ (Devlin et al., 2019)	64.6 (– / –)	65.0 (71.1 / 62.3)
BERT$_{large}$ (Devlin et al., 2019)	72.7 (76.7 / 71.0)	72.0 (76.6 / 70.1)
XLNet$_{large}$ (Yang et al., 2019)	80.1 (– / –)	81.8 (85.5 / 80.2)
XLNet$_{large}$ + DCMN+ (Zhang et al., 2020a)	– (– / –)	82.8 (86.5 / 81.3)
RoBERTa$_{large}$ (Liu et al., 2019)	– (– / –)	83.2 (86.5 / 81.8)
RoBERTa$_{large}$ + MMM (Jin et al., 2020)	– (– / –)	85.0 (89.1 / 83.3)
T5-11B (Raffel et al., 2020)	– (– / –)	87.1 (– / –)
ALBERT$_{xxlarge}$ + DUMA (Zhu et al., 2020)	88.1 (– / –)	88.0 (90.9 / 86.7)
T5-11B + UnifiedQA (Khashabi et al., 2020)	– (– / –)	89.4 (– / –)
Megatron-BERT-3.9B (Shoeybi et al., 2019)	– (– / –)	89.5 (91.8 / 88.6)
ALBERT$_{xxlarge}$ + SC + TL (Jiang et al., 2020)	– (– / –)	90.7 (92.8 / 89.8)
ALBERT$_{base}$ (rerun)	67.9 (72.3 / 65.7)	67.2 (72.1 / 65.2)
POI-Net on ALBERT$_{base}$	72.4 (76.3 / 70.0)	71.0 (75.7 / 69.0)
ALBERT$_{xxlarge}$ (rerun)	86.6 (89.4 / 85.2)	86.5 (89.2 / 85.4)
POI-Net on ALBERT$_{xxlarge}$	88.1 (91.3 / 86.3)	88.3 (91.5 / 86.8)

Table 10: Public submissions on RACE. The results in the first domain are from the leaderboard. SC denotes single choice and TL denotes transfer learning.

Model	**EM**	**F1**
SAN (Liu et al., 2017)	76.2	84.1
R.M-Reader (Hu et al., 2018)	81.2	87.9
ALBERT$_{base}$ (Lan et al., 2020)	82.9	89.3
BERT$_{base}$ (Devlin et al., 2019)	80.8	88.5
BERT$_{large}$ (Devlin et al., 2019)	85.5	92.2
ALBERT$_{xxlarge}$ (Lan et al., 2020)	88.3	94.1
SpanBERT* (Joshi et al., 2020)	88.8	94.6
XLNet$_{large}$ (Yang et al., 2019)	89.7	95.1
RoBERTa$_{large}$ + LUKE (Yamada et al., 2020)	89.8	95.0
ALBERT$_{base}$ (rerun)	82.7	89.9
POI-Net on ALBERT$_{base}$	84.5	91.3
ALBERT$_{xxlarge}$ (rerun)	88.2	94.1
POI-Net on ALBERT$_{xxlarge}$	89.5	95.0

Table 11: Comparison works on SQuAD 1.1 development set. Results with * are from (Clark et al., 2020).

Model	**EM**	**F1**
ALBERT$_{base}$ (Lan et al., 2020)	77.1	80.0
BERT$_{base}$ (Devlin et al., 2019)	77.6	80.4
NeurQuRI (Back et al., 2020)	80.0	83.1
BERT$_{large}$ (Devlin et al., 2019)	82.2	85.0
SemBERT (Zhang et al., 2020b)	84.2	87.9
ALBERT$_{xxlarge}$ (Lan et al., 2020)	85.1	88.1
SpanBERT* (Joshi et al., 2020)	85.7	88.7
XLNet$_{large}$ (Yang et al., 2019)	87.9	90.6
ELECTRA (Clark et al., 2020)	88.0	90.6
ALBERT$_{xxlarge}$ + Retro-Reader (Zhang et al., 2021)	87.8	90.9
ALBERT$_{base}$ (rerun)	77.3	80.4
POI-Net on ALBERT$_{base}$	79.8	82.9
ALBERT$_{xxlarge}$ (rerun)	85.4	88.5
POI-Net on ALBERT$_{xxlarge}$	87.7	90.6

Table 12: Comparison works on SQuAD 2.0 development set. Results with * are from (Clark et al., 2020).