Multi-range Reasoning for Machine Comprehension

Yi Tay[1], **Luu Anh Tuan**[2], **and Siu Cheung Hui**[3]

[1,3]Nanyang Technological University

[2]Institute for Infocomm Research

ytay017@e.ntu.edu.sg[1]

at.luu@i2r.a-star.edu.sg[2]

asschui@ntu.edu.sg[3]

Abstract

We propose MRU (Multi-Range Reasoning Units), a new fast compositional encoder for machine comprehension (MC). Our proposed MRU encoders are characterized by multi-ranged gating, executing a series of parameterized contract-and-expand layers for learning gating vectors that benefit from long and short-term dependencies. The aims of our approach are as follows: (1) learning representations that are concurrently aware of long and short-term context, (2) modeling relationships between intra-document blocks and (3) fast and efficient sequence encoding. We show that our proposed encoder demonstrates promising results both as a standalone encoder and as well as a complementary building block. We conduct extensive experiments on three challenging MC datasets, namely RACE, SearchQA and NarrativeQA, achieving highly competitive performance on all. On the RACE benchmark, our model outperforms DFN (Dynamic Fusion Networks) by $1.5\% - 6\%$ without using any recurrent or convolution layers. Similarly, we achieve competitive performance relative to AMANDA [17] on the SearchQA benchmark and BiDAF [23] on the NarrativeQA benchmark without using any LSTM/GRU layers. Finally, incorporating MRU encoders with standard BiLSTM architectures further improves performance, achieving state-of-the-art results.

1 Introduction

Teaching machines to read, comprehend and reason lives at the heart of machine comprehension (MC) tasks [22, 18, 8, 16]. In these tasks, the goal is to answer questions based on a given passage, effectively testing the learner's capability to understand natural language. This has been an extremely productive area of research in the recent years, giving rise to many highly advanced neural network architectures [36, 23, 11, 25, 31]. A common denominator in many of these models is the compositional encoder, i.e., usually a bidirectional recurrent-based (LSTM [10] or GRU [5]) encoder that sequentially parses the text sequence word-by-word. This helps to model compositionality of words, capturing rich and complex linguistic and syntactic structure in language.

While the usage of recurrent encoder is often regarded as indispensable in highly complex MC tasks, there are still several challenges and problems pertaining to it's usage in modern MC tasks. Firstly, documents can be extremely long to the point where running a BiRNN model across a long document is computationally prohibitive. This is aggravated since MC tasks can be easily extended to reasoning over multiple long documents. Secondly, recurrent encoders have limited access to long term context since each word is sequentially parsed. This restricts any form of multi-sentence and intra-document reasoning from happening within compositional encoder layer.

To this end, we propose a new compositional encoder that can either be used in-place of standard RNN encoders or serve as a new module that is complementary to existing neural architectures. Our proposed MRU encoders learns gating vectors via multiple contract-and-expand layers at multiple

dilated resolutions. Specifically, we compress the input document an arbitrary k times at multi-ranges (e.g., $1, 2, 4, 10, 25$) into a neural bag-of-words (summed) representation. The compact sequence is then passed through affine transformation layers and then re-expanded to the original sequence length. The k document representations (at multiple ranges and n-gram blocks) are then combined and modeled with fully connected layers to form the final compositional gate which are applied onto the original input document. This can be interpreted as compositional gating by exploiting information at multiple-ranges, modeling relationships across different granularities and hierarchies. Intuitively, this is because 1-gram blocks are compared with 2-gram blocks and 10-gram blocks and so on.

This has several advantages. Firstly, we enable a major speedup by avoiding either costly step-by-step gate construction while still maintaining interactions between neighboring words. As such, our model belongs to a class of architectures which is inspired by QRNNs [2] and SRUs [19]. The key difference is that our gates are not constructed by convolution layers but explicit block-based matching across multiple ranges. Secondly, modeling at a long range (e.g., 25 or 50) enables our model to look further ahead as opposed to only one step forward. As such, the learned gastes possess not only information about nearby words but also a larger overview of the context. This is in similar spirit to self-attention, albeit executing within the encoder. Thirdly, the final gates are formed by modeling relationships between multi-range projections (n-gram blocks), allowing for fine-grained intra-document relationships to be captured. The overall contributions of our work is as follows:

- We propose MRU (**M**ulti-range **R**easoning **U**nits), a new compositional encoder which construct gates from a novel contract-and-expand operation. We propose an overall architecture that utilizes MRU within a bi-attentive framework for both multiple choice and span prediction MC tasks. MRU can be used as a standalone (without RNNs) for fast reading and/or together with RNN models (i.e., MRU-LSTM) for more expressive reading.

- We conduct extensive experiments on three large-scale and challenging machine comprehension datasets - RACE [18], SearchQA [8] and NarrativeQA [16]. Our model is lightweight, fast and efficient, achieving state-of-the-art or highly competitive performance on all benchmarked datasets. Since MC datasets often require a considerable amount of reasoning and natural language understanding, we believe that they serve as good testbeds for benchmarking encoders.

- On RACE, our model outperforms Dynamic Fusion Networks (DFN) [37], a highly complex model. While DFN takes approximately a week to train, spending at least several hours per epoch, our model converges in less than 12 hours with only $4-5$ minutes per epoch. Moreover, our model outperforms DFN by $2\% - 6\%$ on the RACE benchmark and other strong baselines such as the Gated Attention Reader by 10%. On RACE, we outperform DFN without any recurrent and convolution layers. Ablation studies show an improvement of up to 6% when using MRU over a LSTM/GRU encoder.

- On the recent SearchQA benchmark [8], we achieve competitive performance relative to AMANDA [17], a state-of-the-art model without using any recurrent or convolution layers. Our model runs at 2 minutes per epoch, approximately five times faster than AMANDA. Incorporating our MRU block with standard BiLSTM architectures (MRU-LSTM) outperforms AMANDA by a reasonable margin.

- On the NarrativeQA benchmark (summaries setting) [16], our MRU encoders achieves highly competitive performance relative to BiDAF [23] a strong MC baseline without using any LSTM/GRU layers. On the other hand, MRU-LSTM significantly outperforms BiDAF, achieving state-of-the-art performance on this dataset.

2 Our Proposed MRU Encoder

In this section, we describe our proposed MRU encoder. The inputs to the MRU encoder is an input document $\{w_1, w_2 \cdots w_\ell\}$, and list of ranges $\{r_1, r_2 \cdots r_k\}$ where k is the number of times the contract and expand operation is executed. The final output of the encoder is a sequence of vectors which retain the same dimensionality as its inputs. Figure 1 (left most block) provides an illustration of the overall encoder architecture.

2.1 Contract-and-Expand Operation

This section describes the operation for each r_j. For the sake of brevity, we drop the superscripts j. For each r_j and the input document, the contract operation performs takes the summation of every r_j words. This reduces the overall document length to ℓ/r_j where each item in the sequence is the sum of every r_j words. Given the new sequence of ℓ/r_i tokens, we then pass each token into a single layered feed-forward neural network:

$$\bar{w}_t = \sigma_r(\mathbf{W}_a(w_t)) + \mathbf{b}_a \tag{1}$$

where $\mathbf{W}_a \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_a \in \mathbb{R}^d$ are the parameters of the contract layer. σ_r is the ReLU activation function. w_t is the t-th token in the sequence. Given the transformed tokens $\bar{w}_1, \bar{w}_2 \cdots \bar{w}_{\ell/r_j}$, we then expand them into the original sequence length. Note that for each r_j, the parameters $\mathbf{W}_a, \mathbf{b}_a$ are not shared.

2.2 Reasoning over Multi-ranged Blocks

From k different calls of the Contract-and-Expand operation at different ranges, we pass the concatenated vector of all transformed tokens into a two layered feed-forward neural network.

$$g_t = \mathbf{F}_2(\mathbf{F}_1([w_t^1; w_t^2; \cdots w_t^k])) \tag{2}$$

where $\mathbf{F}_1(.), \mathbf{F}_2(.)$ are feed-forward networks with ReLU activations, i.e., $\sigma_r(W_x + b)$. $[;]$ is the concatenation operator. g_t is interpreted as a gating vector learned from multiple ranges and Equation (2) is learning the relationships between a token's representation at multiple hierarchies depending on the values of r_j. Notably, it is easy to see that every n pairs of words will have the same gating vector where n is the lowest value of r_j. As such, the value of the 1gram, i.e., $r_j = 1$ (projection of every single token) is critical as it prevents identical gating vectors across the sequence.

2.3 MRU Encoding Operation

To learn the MRU encoded representation of each word, we consider two variations of MRU encoders.

2.3.1 Simple MRU

In this variation, we use g_t as a gating vector to control the fine-grained balanced between the projection of each word w_t in the original input document and the original representation.

$$z_t = tanh(\mathbf{W}_p \, w_t) + \mathbf{b}_p \tag{3}$$
$$y_t = \sigma(g_t) * w_t + (1 - \sigma(g_t)) \, z_t \tag{4}$$

where $\{y_1, y_2, \cdots y_\ell\}$ is the output document representation. σ is the sigmoid function. Note that this formulation is in similar spirit to highway networks [26]. However, since our gating function is learned via multi-range reasoning, it captures more compositionality and long range context. Note that an optional and additional projection may be applied to w_t but we found that it did not yield much empirical benefit.

2.3.2 Recurrent MRU

In the second variation, we consider a recurrent (sequential) variant. This is in similar spirit to QRNNs [2] and SRUs [19] which reduces computation cost by pre-learning the gating vectors. The following operations describe the operations of the recurrent MRU cell for each timestep t.

$$c_t = g_t \odot c_{t-1} + (1 - g_t) \odot z_t \tag{5}$$
$$h_t = o_t \odot c_t \tag{6}$$

where c_t, h_t are the cell and hidden states at time step t. g_t are the gates learned from out multi-range reasoning step. o_t is an additional output gate learned via applying an affine transform on the input vector w_t, i.e., $o_t = W_o(w_t) + b_o$. Similar to RNNs, the Recurrent MRU parses the input sequence word-by-word. However, the cost is significantly reduced because we do not have expensive matrix operations that are executed in a non-parallel fashion. Finally, the outputs of the MRU encoder are a series of hidden vectors $\{h_1, h_2 \cdots h_\ell\}$ for each word in the sequence.
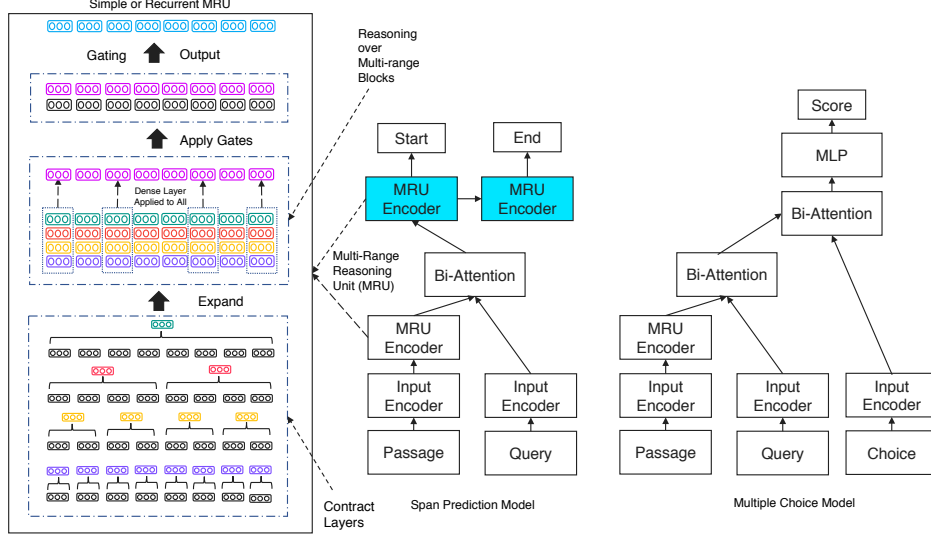
3

Figure 1: High-level overview of our proposed MRU encoder (left) and Bi-Attentive architecture for two types of MC tasks. MRU illustration shows contraction values of $\{1, 2, 4, 8\}$. Documents are contracted and then projected with affine transformations. Subsequently, they are expanded to the original document length. A new projection layer compares the representations across multiple ranges. This multi-range representation is used as a gate to influence the input sequence.

3 Overall Model Architectures

This section describes the overall model architecture that utilizes MRU encoders. In our experiments, we focus on both multiple-choice based (RACE) and span prediction MC tasks (SearchQA, NarrativeQA). Since the core focus of this paper is our encoder, we briefly provide the high-level details of our vanilla Bi-Attentive model. The Bi-Attentive models that are used in our experiments act as baselines, often being less complex than current competitive models such as BiDAF [23], AMANDA [17] or DFN [37].

3.1 Multiple Choice Models

In MCQ models, there are three types of input sequences, namely Passage (P), Question (Q) and Answers (A_j). The output of the model (for each answer), is a score $s(P, Q, A_j) \in [0, 1]$ denoting the strength of A_j. The problem is formulated as a listwise approach, in which multiple answers are modeled concurrently with respect to P, Q.

- **Input Encoding** - Each input sequence is passed into first a projection layer. To enhance the input word representations, we also include the standard EM (exact match) binary feature to each word. In this case, we use a three-way EM adaptation, i.e., $EM(P, Q), EM(Q, A)$ and $EM(P, A)$. The projected embeddings are then passed into a single-layered highway network.

- **Compositional Encoder** - In our experiments, we vary the encoder in this layer. Typical choices of encoders in this layer are LSTMs or GRUs. We vary this in our experiments in order to benchmark the effectiveness of our proposed MRU encoder. The output of this layer is same dimensions as its inputs (typically the hidden states of a RNN model).

- **Bi-Attention Layer** - This layer models the interactions between P, Q and A. Let $B(.)$ be a standard bidirectional attention that utilizes mean-pooling aggregation. The scoring function is the bilinear product of the nonlinearly transformed input i.e., $F(x)_i^\top \mathbf{M} F(y)_i$. We first apply $B(P, Q)$ to form bi-attentive P^q, Q^p representations. Subsequently, we apply $B(P^q, A_j)$ to learn a vector representation for each answer. A temporal sum pooling is applied on the outputs of P^{qa}, A_j^p and concatenated to form $a_j^f \in \mathbb{R}^{2d}$.

4

- **Answer Selection** Let $\{a_1, a_2 \cdots a_{N_a}\}$ be the inputs to this layer and N_a is the number of answer candidates. Motivated by work in retrieval-based QA [24, 27], we include word overlap features to each answer candidate. This word overlap feature is in similar spirit to the EM feature. Each overlap operation between two sequence returns four features. We convert each answer vector a_j into a scalar via $a_j^f = Softmax(\mathbf{W}_2(\sigma_r(\mathbf{W}_1([a_j]) + b_1) + b_2))$.

The MCQ-based model minimizes the multi-class cross entropy where the number of classes corresponds to the number of choices.

3.2 Span Prediction Model

Span prediction models models the relationship between P and Q. The goal is to extract (or predict a span s, e) where $P[s : e]$ is the answer to the query. For most part, the model architecture remains similar especially for the input encoding layers compositional encoder layer. The key difference is that we reduce the number of input sequence from three to two.

- **Input Encoding** - This follows the same design as the MCQ model, albeit for two sequence. Similarly, the two-way EM feature is added before passing into the highway layer.
- **Compositional Encoder** - This remains identical as the MCQ-based model.
- **Bi-Attention Layer** - We adopt a different bi-attention function for span prediction. More specifically, we use the 'SubMultNN' or the ''Mult' adaptation from [30] (this is tuned) and compare aligned sequences between P and Q to form P^q, the query-dependent passage representation.
- **Answer Pointer Layer** - In this layer, we pass P^q through a two layered compositional encoder (which is varied). The start pointer and end pointer is determined by $F(H_1), F(H_2)$ where H_1, H_2 are the hidden outputs from the first and second encoder respectively. $F(.)$ is a linear transform, projecting each hidden state to a scalar. We pass both of them into softmax functions to obtain probability distributions.

Following [23, 31], we minimize the joint cross entropy loss of the start and end probability distributions. During inference, finding the best answer span follows [31].

4 Empirical Evaluation

In this section, we report our experimental results and comparisons against other published work.

4.1 Datasets

For our experiments, we use one challenging multiple choice MC dataset and two span-prediction MC datasets.

- **RACE** (Reading Comprehension from Examinations) [18] is a recently proposed dataset that is constructed from real world examinations. Given a passage, there are several questions with four options each. The authors argue that RACE is more challenging compared to popular benchmarks (e.g., SQuAD [22]) as more multi-sentence and compositional reasoning is required. There are two subsets of RACE, namely RACE-M (Middle school) and RACE-H (High school).
- **SearchQA** [8] is a recent dataset that emulates a real world QA system. It involves extracting passages from search engine results and require models to answer questions by reasoning and reading these search snippets.
- **NarrativeQA** [16] is a recent benchmark proposed for story-based reading comprehension. Different from many MC datasets, the answers are handwritten by human annotators.

MCQ datasets are evaluated using the standard accuracy metric. For RACE, we train models on the entire dataset, i.e., both RACE-M and RACE-H and evaluate separately. For RACE, the model selection is based on each subset's respective development set. For SearchQA, we follow [17, 8] which evaluates unigram exact match (EM) and n-gram F1 scores. For NarrativeQA, since the

Model	RACE-M	RACE-H	RACE	Time
Sliding Window [18]	37.3	30.4	32.2	N/A
Stanford AR [4]	44.2	43.0	43.3	N/A
GA [7]	43.7	44.2	44.1	N/A
ElimiNet [20]	N/A	N/A	44.5	N/A
Dynamic Fusion Network [37]	51.5	45.7	47.4	$\approx$8 hours (1 week*)
BiAttention (No Encoder)	50.6	44.0	44.9	3 min (9 hours)
BiAttention ($250d$ GRU)	48.5	42.1	44.0	16 min (2 days)
BiAttention ($250d$ LSTM)	50.3	40.9	43.6	18 min (2 days)
BiAttention ($250d$ Sim. MRU)	**57.7**	<u>47.4</u>	**50.4**	4 min (12 hours)
BiAttention ($250d$ MRU)	<u>56.1</u>	**47.5**	<u>50.0</u>	12 min (20 hours)
GA + ElimiNet [20]	N/A	N/A	47.2	N/A
DFN Ensemble (x9) [37]	55.6	49.4	51.2	N/A
BiAttention (MRU) Ensemble (x9)	**60.2**	**50.3**	**53.3**	N/A

Table 1: Comparison against other published models on RACE dataset [18]. Competitor result are reported from [18, 37]. Best result for each category (single and ensemble) is in boldface. Last column reports estimated training time per epoch and total time for convergence. * estimated values that we obtain from asking the authors.

answers are human written and not constrained to spans in the passage, the evaluation metrics are Bleu-1, Bleu-4, Meteor and Rouge-L following [16].

4.2 Competitor Methods

We describe the key competitors on each dataset.

- **RACE** - the key competitors are the Stanford Attention Reader (Stanford AR) [4], Gated Attention Reader (GA) [7], and Dynamic Fusion Networks (DFN) [37]. GA incorporates a multi-hop attention mechanism that helps to refine the answer representations. DFN is an extremely complex model. It uses BiMPM's matching functions [34] for extensive matching between Q, P and A, multi-hop reasoning powered by ReasoNet [25] and employs reinforcement learning techniques for dynamic strategy selection.

- **SearchQA** - the main competitor baseline is the AMANDA model proposed by [17]. AMANDA uses a multi-factor self-attention module, along with a question focused span prediction. AMANDA also uses BiLSTM layers for input encoding and at the span prediction layers. We also compare against the reported ASR [13] baselines which was reported in [8].

- **NarrativeQA** - On the NarrativeQA benchmark, we compare with the reported baselines in [16]. We compete on the summaries setting, in which the baselines are a context-less sequence to sequence (seq2seq) model, ASR [13] and BiDAF [23].

4.3 Our Methods

Across our experiments, we benchmark several variants of our proposed MRU. The first is denoted as Sim. MRU which corresponds to the Simple MRU model described earlier. The model denoted by MRU (without any prefix) corresponds to the recurrent MRU model. Finally, the final variant is the MRU-LSTM which places a MRU encoder layer on top of a BiLSTM layer. We report the dimensions of the encoder as well as training time (per epoch) for each variant. The encompassing framework for MRU is the Bi-Attentive models described for MCQ-based problems and Span prediction problems. Unless stated otherwise, the encoder in the pointer layer for span prediction models also uses MRU. However, for the Hybrid MRU-LSTM models, answer pointer layers use BiLSTMs. For the RACE-dataset, we additionally report scores of an ensemble of nine Sim. MRU models. This is to facilitate comparison against ensemble models of [37].

4.4 Implementation Details

We implement all models in TensorFlow [1]. Word embeddings are initialized with $300d$ GloVe [21] vectors and are not fine-tuned during training. Dropout rate is tuned amongst $\{0.1, 0.2, 0.3\}$ on all layers including the embedding layer. For our MRU model, we use a range values of $\{1, 2, 4, 10, 25\}$.

	Dev		Test		
Model	Acc	F1	Acc	F1	Time
TF-IDF max [8]	13.0	N/A	12.7	N/A	N/A
ASR [13]	43.9	24.2	41.3	22.8	N/A
AMANDA [17]	48.6	57.7	46.8	56.6	$\approx 8^*$ min
Bi-Attention[†] (No Encoder)	12.4	20.2	18.9	12.3	≈ 17 sec
Bi-Attention[†] ($150d$ BiLSTM)	40.0	51.3	38.6	49.0	≈ 7 min
Bi-Attention[†] ($300d$ LSTM)	40.3	48.7	38.2	46.4	≈ 6 min
Bi-Attention[†] ($300d$ Sim. MRU)	44.1	45.5	42.9	43.1	≈ 25 sec
Bi-Attention[†] ($300d$ MRU)	48.6	54.8	46.8	53.3	≈ 2 min
Bi-Attention ($200d$ Hybrid MRU-LSTM)	**50.5**	**59.9**	**49.4**	**59.5**	≈ 7 min

Table 2: Experimental Results on SearchQA dataset. [8]. Unigram Accuracy and N-gram F1 are reported following [17]. All models with [†] use the same encoder in the answer pointer layer. [*] are estimates running a replicated model with same batch size ($b = 256$) as our models.

MRU encoders are only applied on the passage and not the query. We adopt the Adam optimizer [15] with a learning rate of $0.0003/0.001/0.001$ for RACE/SearchQA/NarrativeQA respectively. The batch size is set to $64/256/32$ accordingly. The maximum sequence lengths are $500/200/1100$ respectively. For NarrativeQA, we use the Rouge-L score to find the best approximate answer relative to the human written answer for training the span model. All models are trained and all runtime benchmarks are based on a TitanXP GPU.

4.5 Experimental Results on RACE

Table 1 reports our results on the RACE benchmark dataset. Our proposed MRU model achieves the best result for both single models and ensemble models. We outperform highly complex models such as DFN. We also pull ahead of other recent baselines such as ElimiNet and GA by at least 5%. The best single model score from RACE-H and RACE-M alternates between Sim. MRU and MRU. Overall, there is a 6% improvement on the RACE-H dataset and 1.8% improvement on the RACE-M dataset. Our Sim. MRU model also runs at 4 min per iteration, which is dramatically faster and simpler than DFN or other recurrent models. We believe that this finding highlights the importance of designing strong and fast baselines for the task at hand.

In general, we also found that the usage of a recurrent cell is not really crucial on this dataset since (1) Sim. MRU and MRU can achieve comparable performance to each other, (2) GRU and LSTM models do not have a competitive edge and (3) Using no encoder already achieves comparable[1] performance to DFN. Finally, an ensemble of Sim. MRU models achieve state-of-the-art performance on the RACE dataset, achieving and overall score of 53.3%.

4.6 Experimental Results on SearchQA

Table 2 reports our results on the SearchQA dataset. We draw the reader's attention to the performance of the $300d$ MRU encoder. We achieve the same accuracy as AMANDA without using any LSTM or GRU encoder. This model runs at 2 min per epoch, making it 4 times more efficient than AMANDA (estimated, with identical batch size). While, AMANDA also uses multi-factor self-attention, along with character enhanced representations, our simple MRU encoder used within a mere baseline bi-attentive framework comes close in performance. Finally, the hybrid combination, MRU-LSTM significantly outperforms AMANDA by 3%.

Contrary to MCQ-based datasets, we found that Sim. MRU model could not achieve comparable results to the recurrent MRU. We hypothesize that this is due to the need to predict spans. Nevertheless, the $300d$ MRU outperforms an LSTM encoder and remain competitive to a BiLSTM of similar dimensionality. We also observe that LSTM and MRU are complementary. This is made evident by how stacking MRUs over LSTMs can give a performance boost relative to using each encoder separately.

---

[1]Nevertheless, this suggests the importance of benchmarking good and strong baselines since a well-tuned baseline model can outperform DFN, a highly complicated model.

Model	Bleu-1	Bleu-4	Meteor	Rouge-L	Time
Seq2Seq[†]	15.89	1.26	4.08	13.15	N/A
ASR[†] [13]	23.20	6.39	7.77	22.26	N/A
BiDAF[†] [23]	33.72	15.53	15.38	36.30	N/A
BiAttention ($300d$ LSTM)	31.18	15.34	14.42	32.95	$\approx$1 hour
BiAttention ($150d$ BiLSTM)	34.22	18.22	16.19	38.32	$\approx$1 hour
BiAttention ($300d$ Sim. MRU)	9.15	1.69	3.95	11.16	1 min
BiAttention ($300d$ MRU)	33.28	16.15	15.84	36.65	18 mins
BiAttention ($150d$ Hybrid MRU-LSTM)	**36.55**	**19.79**	**17.87**	**41.44**	$\approx$1 hour

Table 3: Experimental Results on the NarrativeQA reading comprehension challenge [16] using summaries. [†] are baselines reported by [16].

4.7 Experimental Results on NarrativeQA

Table 3 reports our results on the NarrativeQA benchmark. First, we observe that $300d$ MRU can achieve comparable performance with BiDAF [23]. When compared with a BiLSTM of equal output dimensions ($150d$), we find that our MRU model performs competitively, with less than 1% deprovement across all metrics. However, the time cost required is significantly reduced. The performance of our model is significantly better than $300d$ LSTM model while also being significantly faster. Here, we note that Sim. MRU does not produce reasonable results at all, which seems to be in similar vein to results on SearchQA, i.e., a recursive cell that processes word-by-word is mandatory for span prediction. However, our results show that it is not necessary to construct gates in a word-by-word fashion. Finally, the MRU-LSTM significantly outperforms all models, including BiDAF on this dataset. Performance improvement over the vanilla BiLSTM model ranges from $1\% - 3\%$ across all metrics, suggesting that MRU encoders are also effective as a complementary neural building block.

5 Related Work

A diverse collection of MC datasets such as SQuAD [22] and CNN/DailyMail [9] are readily available for benchmarking new deep learning models. New datasets have been recently released [16, 12, 18, 35], claiming to involve a greater need for going beyond simple surface-level matching. As such, these datasets often emphasize the extent of compositional and multi-sentence reasoning required to tackle its questions. In the recent years, a wide range of innovation solutions have also been proposed, mainly involving bi-attention [23, 36, 6] and answer pointers [31]. Recent work also investigates the notion of multi-hop reasoning [7, 25, 37], reinforcement learning [25, 32, 11] and self-matching / self-attention [17, 33]. While many of these works use BiLSTMs are standard building blocks, recent work [38] attempts a RNN-less model architecture by utilizing components inspired by the Transformer architecture [29]. Our work is mainly concerned with designing an efficient encoder that is able to capture not only compositional information but also long-range and short-range information. More specifically, our recurrent MRU encoder takes on a similar architecture to Quasi-Recurrent Neural Networks [2] and Simple Recurrent Units [19]. A recent work, Cross Temporal Recurrent Networks [28] extends QRNNs by fusing temporal gates across question-answer pairs. In these models, gates are pre-learned and then applied. However, different from existing models such as QRNNs that convolution layers as gates, we use a block-based contract-and-expand layers for learning gates. Finally, our model also draws inspiration from dilation, in particular dilated RNNs [3] and dilated convolutions [14], that intuitively help to model long-range dependencies.

6 Conclusion and Future Work

We proposed a novel neural architecture, the MRU encoder and an overall bi-attentive model for both MCQ-based and span prediction MC tasks. We apply it to three MC datasets and achieve competitive performance on all without the use of recurrent layers. Our proposed method outperforms DFN, an extremely complex model, without using any LSTM or GRU layer. We also remain competitive to AMANDA and BiDAF without any LSTM/GRU. While our proposed encoder demonstrates promise on reasoning and understanding natural language, we believe that our encoder is generalizable to other domains beyond machine comprehension. However, we defer this prospect to future work.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-recurrent neural networks. *CoRR*, abs/1611.01576, 2016.

[3] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. Dilated recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 76–86, 2017.

[4] Danqi Chen, Jason Bolton, and Christopher D Manning. A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*, 2016.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[6] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423*, 2016.

[7] Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*, 2016.

[8] Matthew Dunn, Levent Sagun, Mike Higgins, Ugur Guney, Volkan Cirik, and Kyunghyun Cho. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*, 2017.

[9] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Minghao Hu, Yuxing Peng, and Xipeng Qiu. Mnemonic reader for machine comprehension. *arXiv preprint arXiv:1705.02798*, 2017.

[12] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[13] Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. Text understanding with the attention sum reader network. *arXiv preprint arXiv:1603.01547*, 2016.

[14] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[16] Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The narrativeqa reading comprehension challenge. *arXiv preprint arXiv:1712.07040*, 2017.

[17] Souvik Kundu and Hwee Tou Ng. A question-focused multi-factor attention network for question answering. February 2018.

[18] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[19] Tao Lei and Yu Zhang. Training rnns as fast as cnns. *arXiv preprint arXiv:1709.02755*, 2017.

[20] Soham Parikh, Ananya Sai, Preksha Nema, and Mitesh M Khapra. Eliminet: A model for eliminating options for reading comprehension with multiple choice questions, 2018.

[21] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014.

[22] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[23] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[24] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 373–382, 2015.

[25] Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055. ACM, 2017.

[26] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015.

[27] Yi Tay, Minh C. Phan, Anh Tuan Luu, and Siu Cheung Hui. Learning to rank question answer pairs with holographic dual LSTM architecture. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 695–704, 2017.

[28] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Cross temporal recurrent networks for ranking question answer pairs. *arXiv preprint arXiv:1711.07656*, 2017.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010, 2017.

[30] Shuohang Wang and Jing Jiang. A compare-aggregate model for matching text sequences. *CoRR*, abs/1611.01747, 2016.

[31] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*, 2016.

[32] Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. R3: Reinforced reader-ranker for open-domain question answering. *arXiv preprint arXiv:1709.00023*, 2017.

[33] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 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)*, volume 1, pages 189–198, 2017.

[34] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4144–4150, 2017.

[35] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *arXiv preprint arXiv:1710.06481*, 2017.

[36] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *CoRR*, abs/1611.01604, 2016.

[37] Yichong Xu, Jingjing Liu, Jianfeng Gao, Yelong Shen, and Xiaodong Liu. Towards human-level machine reading comprehension: Reasoning and inference with multiple strategies. *arXiv preprint arXiv:1711.04964*, 2017.

[38] Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*, 2018.