

MEM2MEM: LEARNING TO SUMMARIZE LONG TEXTS WITH MEMORY COMPRESSION AND TRANSFER

Anonymous authors

Paper under double-blind review

ABSTRACT

We introduce *Mem2Mem*, a memory-to-memory mechanism for hierarchical recurrent neural network based encoder decoder architectures and we explore its use for abstractive document summarization. Mem2Mem transfers “memories” via readable/writable external memory modules that augment both the encoder and decoder. Our memory regularization compresses an encoded input article into a more compact set of sentence representations. Most importantly, the memory compression step performs implicit extraction without labels, sidestepping issues with suboptimal ground-truth data and exposure bias of hybrid extractive-abstractive summarization techniques. By allowing the decoder to read/write over the encoded input memory, the model learns to read salient information about the input article while keeping track of what has been generated. Our Mem2Mem approach yields results that are competitive with state of the art transformer based summarization methods, but with *16 times fewer parameters*.

1 INTRODUCTION

Automatic summarization is the automated process of reducing the size of an input text while preserving its most relevant information content and its core semantics. Techniques for summarization are often characterized as being either: *Extractive* or *Abstractive*. *Extractive* methods construct summaries by combining the most salient passages (usually whole sentences) of a source text; a process similar to human’s way of identifying the right information. One way to achieve extractive summarization is to define the problem as a sentence classification task, using some form of representation of the sentences in a document (Nallapati et al., 2016b; Cheng & Lapata, 2016). To avoid content overlap issues, previous work has used sentence reranking (Narayan et al., 2018) or sentence ordering by extracting sentences recurrently (Chen & Bansal, 2018). *Abstractive* methods generate summaries by generating new sentence constructs “from scratch”, or from representation of document content, a process that is conceptually more similar to the notion of paraphrasing. Abstractive text summarization has attracted interest since it is capable of generating novel formulations of summaries using language generation models conditioned on the source text. Several attention-based Recurrent Neural Network (RNN) encoder-decoders have been introduced to tackle varying text generation issues of standalone abstractive sequence-to-sequence (seq2seq) models. Copy and pointer mechanisms (Gu et al., 2016; Tu et al., 2016; Vinyals et al., 2015; See et al., 2017), for example, have enabled decoders to better generate unseen words, out-of-vocabulary words and named entities.

Most recently, hybrid extractive and abstractive architectures have been proposed and have shown promising results in both quantitative performance measures and human evaluations. In such setups, the extractive model first selects salient sentences from a source article, and the abstractive model paraphrases the extracted sentences into a final summary. The majority of current state-of-the-art abstractive summarization models¹ are based on the hybrid approach (Chen & Bansal, 2018; Gehrmann et al., 2018; Hsu et al., 2018; Bae et al., 2019; Subramanian et al., 2019). Nonetheless, hybrid models can be limited by three disadvantages. First, since ground-truth labels for extractive summarization are usually not provided, extractive labels must be generated by a potentially suboptimal algorithm (Hsu et al., 2018; Subramanian et al., 2019). The performance of models trained

¹Excluding summarization models using large scale pre-trained language models such as BERT (Devlin et al., 2019)

with such labels is therefore bounded by the quality of the performance of the extractive heuristics. Second, since ground-truth binary labels for recurrently extracted sentences are typically teacher forced as in Chen & Bansal (2018), “exposure bias” (Ranzato et al., 2016) may negatively affect content selection performance at inference. Finally, given that the hard extraction step is not differentiable, existing hybrid models typically require multi-step training (not end-to-end) (Gehrmann et al., 2018; Subramanian et al., 2019) or reinforcement learning (Chen & Bansal, 2018; Bae et al., 2019) to train the whole model.

In this paper, we introduce a novel abstractive summarization model that incorporates an intermediate extractive step but does not require labels for this type of extractive content selection, and it is fully end-to-end trainable. To achieve this, we propose a new memory augmented encoder-decoder (MAED) architecture (Wang et al., 2016; Yogatama et al., 2018; Ma & Principe, 2019; Le et al., 2018) called Mem2Mem. Mem2Mem has 2 memorization modes: (1) absorb key information of the encoded source sequence via a compression mechanism, and (2) sequentially update the external memory during target summary generation. Without using extractive ground-truth labels, we find in our analysis that Mem2Mem’s compression mechanism behaves as an implicit sentence extractor that stores sentence representations of the salient content. The choice of sentence representations is only guided by the memory regularization and conditional language modeling loss of the decoder, thus avoiding exposure bias from maximizing the likelihood of sequential binary extraction labels. Finally, the encoded memory is transferred to the decoder memory, which is iteratively refined during the decoding process. To our knowledge, Mem2Mem is the first abstractive summarization model that uses memory compression for sentence extraction and that directly employs the memorized representations during summary generation. We empirically demonstrate the merits of this approach by setting a new state-of-the-art on long text abstractive summarization tasks on the Pubmed, arXiv and Newsroom datasets (Cohan et al., 2018; Grusky et al., 2018). Our contributions are three fold:

1. We introduce the Mem2Mem approach that (i) stores salient sentence-level representations via memory compression, (ii) transfers memory from encoder to decoder and (iii) updates the memory as the summary generation proceeds.
2. Unlike previous works, our model combines the best of extractive and abstractive summarization in a fully end-to-end trainable manner without supervision on the extraction step.
3. Our method yields results that are competitive with state of the art transformer (Vaswani et al., 2017) language model based techniques, but with substantially fewer parameters (only 6% of the parameters used by a recent state of the art transformer based method).

2 BACKGROUND

For our baseline, we use a hierarchical recurrent encoder-decoder structure (HRED) based seq2seq model from Nallapati et al. (2016a) and Cohan et al. (2018). The HRED has two encoder GRUs: a sentence encoder and a document encoder. Given an input sentence of length N , the sentence encoder takes a sequence of token embeddings \mathbf{x} and transforms it into a sequence of hidden states.

$$\mathbf{h}_1^{(w)}, \dots, \mathbf{h}_N^{(w)} = \text{GRU}_{sen}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) \quad (1)$$

The last hidden state of the sentence encoder is used as a corresponding sentence embedding \mathbf{s} . The sequence of sentence embeddings \mathbf{s} are then processed by the document encoder.

$$\mathbf{h}_1^{(s)}, \dots, \mathbf{h}_L^{(s)} = \text{GRU}_{doc}(\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_L) \quad (2)$$

where \mathbf{s}_j is the j th sentence embedding, $\mathbf{h}_j^{(s)}$ is the associated document encoder hidden state and L is the total number of sentences in the document.

The decoder GRU generates a target summary one token at a time. At each decoding step t , the decoder creates a decoder hidden state $\mathbf{h}_t^{(d)}$. The decoder then obtains a context vector \mathbf{c}_t via γ_{ti} , an alignment between $\mathbf{h}_i^{(w)}$ and $\mathbf{h}_t^{(d)}$. γ_{ti} is computed by combining token level attention α and sentence level attention β , such that:

$$\alpha_{ti} = \text{Attn}(\mathbf{h}_t^{(d)}, \mathbf{h}_i^{(w)}), \quad \beta_{tj} = \text{Attn}(\mathbf{h}_t^{(d)}, \mathbf{h}_j^{(s)}), \quad \gamma_{ti} = \frac{\beta_{tm(i)}\alpha_{ti}}{\sum_{l=1}^{N_d} \beta_{tm(l)}\alpha_{tl}} \quad (3)$$

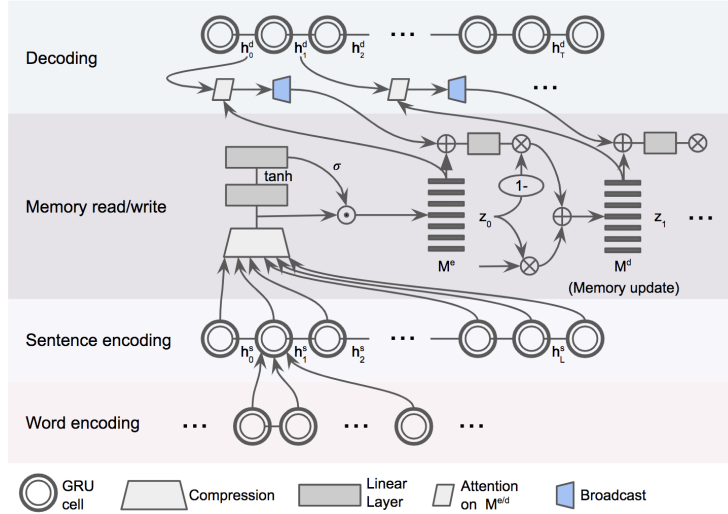


Figure 1: Mem2Mem architecture: Sentence-level representations from the document encoder are reduced to a fix sized encoder memory \mathbf{M}_E . The encoder memory is transferred to the decoder memory \mathbf{M}_D and is read using a RAM like mechanism. The resulting memory readout vector is used to condition sentence and word-level attention. The decoder hidden states and the memory readout vector are then used to update the memory state \mathbf{M}_D via a gated write operation (z_i).

where $m(l)$ denotes the index of the sentence corresponding to the l th word, and N_d is the total number of tokens in the input document. Attn in equation (3) is defined as in Luong et al. (2015).

The probability distribution of the next target word y_t is estimated using the decoder hidden state $\mathbf{h}_t^{(d)}$ and the context vector \mathbf{c}_t . The training objective \mathcal{L} is the average negative log likelihood of the target word y_t over the whole ground truth summary of length T . Finally, the pointer generator and decoder coverage method in See et al. (2017) are used for the baseline HRED. More details of the baseline architecture can be found in Appendix 1.

3 MEMORY-TO-MEMORY MECHANISM (MEM2MEM)

Mem2Mem has three main features. (1) An encoder memory bank that compresses a large set of encoder hidden representations into a smaller subset of salient representations. (2) Read/write operations that allow the decoder to read and update the encoder memory bank with new information during summary generation. (3) Token generation that is conditioned on the extracted sentence representations accessed from the dynamic memory. The Mem2Mem components can be seamlessly integrated with existing HRED architectures. In essence, the whole process can be seen as *extraction* followed by *generation*. The architecture is depicted in Figure 1.

3.1 MEMORY COMPRESSION ON ENCODER

The aim of having an external memory on the encoder is to create a fixed size representation that reduces the set of L hidden representations from GRU_{doc} to a subset of r representations. From a sequence of sentence-level document encoder hidden representations $\mathbf{h}_1^{(s)}, \mathbf{h}_1^{(s)}, \dots, \mathbf{h}_L^{(s)}$, we construct an intermediate 2-D matrix $\mathbf{H} \in \mathbb{R}^{L \times d}$, where d is the document encoder hidden size,

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^{(s)} & \mathbf{h}_2^{(s)} & \dots & \mathbf{h}_L^{(s)} \end{bmatrix}^T. \quad (4)$$

A smaller sized memory bank is generated by taking a linear combination of each row in \mathbf{H} . The weight vector for the linear combination \mathbf{a} is computed with self-attention mechanism.

$$\mathbf{a} = \text{softmax}(w_{a1}^T \tanh(\mathbf{W}_{a2} \mathbf{H}^T)) \quad (5)$$

where $w_{a1} \in \mathbb{R}^{d_a}$, $\mathbf{W}_{a2} \in \mathbb{R}^{d_a \times d}$, and d_a is the size of the hidden layer which is a hyperparameter. To capture various aspects of the input document, \mathbf{a} is extended to a multi-head memory write

attention matrix \mathbf{A} with r heads, $\mathbf{A} = \text{softmax}(\mathbf{W}_{a1}\tanh(\mathbf{W}_{a2}\mathbf{H}^\top))$ where $\mathbf{W}_{a1} \in \mathbb{R}^{r \times d_a}$ and r is a hyperparameter. This results in r different convex combinations of \mathbf{H} , which gives us the final multi-head encoder memory matrix $\mathbf{M}_E \in \mathbb{R}^{r \times d}$, $\mathbf{M}_E = \mathbf{A}\mathbf{H}$. To ensure the attention weights in various heads focus on a diverse set of salient input sentences, we propose a novel regularization loss for memory compression. The following regularization term encourages the diversity of compressed encoded states.

$$\mathcal{L}^{(comp)} = \|\mathbf{A}\mathbf{A}^\top - \mathbf{I}\|_F^2 \quad (6)$$

where $\|\bullet\|_F$ is the Frobenius norm. The regularization loss $\mathcal{L}^{(comp)}$ achieves two goals simultaneously: (1) It promotes diversity over the r sentence representations stored in the memory, thereby reducing the risk of redundant information. (2) It hardens the attention probabilities of each head, assuring that each memory slot is approximately associated to a single sentence representation. As a result, the encoder memory \mathbf{M}_E essentially performs implicit extraction over the encoder hidden states in a fully differentiable manner. Figure 3 in Appendix shows the effect of regularization on the encoder memory compression. Note that no supervision exists on this extractive step and the memory compression is only guided by the memory regularization and back-propagated error signals from the target summary generation.

3.2 READ/WRITE OPERATIONS ON DECODER

Once the encoder memory is constructed, the context read from the memory is used to augment both attention mechanism and the target token generation. As a first step, the encoder memory \mathbf{M}_E is transferred to the decoder and used as an initial state of the decoder memory \mathbf{M}_D . At every time step t , the decoder reads from the memory and generates a memory read vector \mathbf{m}_t . Specifically, the decoder takes the weighted sum over r memory slots via a RAM like attention mechanism:

$$\psi_{tk} = \text{Attn}(\mathbf{h}_t^{(d)}, \mathbf{M}_D(k)), \quad \mathbf{m}_t = \sum_{k=1}^r \psi_{tk} \mathbf{M}_D(k), \quad (7)$$

where $\mathbf{M}_D(k)$ is the vector representation of the k th head or slot of \mathbf{M}_D . $\mathbf{h}_t^{(d)}$ is then combined with \mathbf{m}_t and generate a memory augmented decoder hidden state $\mathbf{h}_t^{(m)} = \mathbf{W}_m[\mathbf{h}_t^{(d)}; \mathbf{m}_t]$ and the next token estimation of the baseline system is replaced with $\mathbf{h}_t^{(m)}$. As a consequence, the attention over the source text and the prediction of the target token are conditioned on the memory read \mathbf{m}_t . Thus, there is a direct link between the contents of the memory and the text generation.

During the summary generation process, the semantics of the source sequence that is kept in the decoder memory needs to be modified. The memory write operation of Mem2Mem enables the memory to log the history of what has been attended and generated. The decoder memory write operation outlined below removes and adds information using a gated mechanism to forget used memories and update each memory slot. The gating mechanism is conditioned on the the memory content \mathbf{M}_D , the memory read vector \mathbf{m}_t , and the decoder hidden state $\mathbf{h}_{t+1}^{(d)}$:

$$\mathbf{z}_t^k = \sigma(\mathbf{W}_{z1}\mathbf{h}_{t+1}^{(d)} + \mathbf{W}_{z2}\mathbf{m}_t + \mathbf{W}_{z3}\mathbf{M}_D(k)) \quad (8)$$

$$\mathbf{u}_t^k = \tanh(\mathbf{W}_{u1}\mathbf{h}_{t+1}^{(d)} + \mathbf{W}_{u2}\mathbf{m}_t + \mathbf{W}_{u3}\mathbf{M}_D(k)) \quad (9)$$

$$\mathbf{M}_D(k) := \mathbf{z}_t^k \odot \mathbf{M}_D(k) + (1 - \mathbf{z}_t^k) \odot \mathbf{u}_t^k \quad (10)$$

Although text generation is directly conditioned on the memory context \mathbf{m}_t , the benefit of having memory was limited in our preliminary experiments. We observed only a few memory slots were repetitively attended during decoding. To ensure that the Mem2Mem decoder fully utilizes its memory to improve generation, we propose another regularization term $\mathcal{L}^{(read)}$.

$$\mathbf{c}_t^{(m)} = \sum_{k=1}^r \psi_{tk} \mathbf{M}_E(k), \quad \mathbf{c}_t^{(s)} = \sum_{j=1}^{L_s} \beta_{tj} \mathbf{h}_j^{(s)}, \quad \mathcal{L}^{(read)} = \frac{1}{T} \sum_{t=1}^T \|\mathbf{c}_t^{(m)} - \mathbf{c}_t^{(s)}\|_2 \quad (11)$$

where $\mathbf{M}_E(k)$ is the vector representation of the k^{th} memory head of \mathbf{M}_E and $\mathbf{c}_t^{(s)}$ is a sentence level context vector. The regularization assumes that if the memory context \mathbf{m}_t , which is a sentence-level representation, is combined properly within the decoding, the sentence level context vector $\mathbf{c}_t^{(s)}$

Table 1: Results on the PubMed, arXiv and Newsroom dataset. Each type corresponds to purely abstractive (A), extractive (E), or extractive-abstractive hybrid (H) approaches. TLM uses the GPT-2 model (Radford et al., 2019) that has 16 times larger parameters than Mem2Mem. The highest ROUGE scores for abstractive methods are boldfaced. All ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE. Results taken from: ¹Subramanian et al. (2019) and ²Cohan et al. (2018). * Newsroom abstractive summarization test set results.

Model	Type	# of params.	ROUGE-1/2/L		
			PubMed	arXiv	Newsroom*
Lead-10 / 3 (Newsroom) ¹	E	-	37.45/14.19/34.07	35.52/10.33/31.44	13.7/2.4/11.2
Sent-CLF ¹	E	-	45.01/19.91/41.16	34.01/8.71/30.31	15.4/2.7/12.8
Sent-PTR ¹	E	-	43.30/17.92/39.47	42.32/15.63/38.06	15.9/2.8/13.0
Attn-Seq2Seq ^{1,2}	A	-	31.55/8.52/27.38	29.3/6.00/25.56	6.21/1.07/5.68
Ptr-Gen-Seq2Seq ^{1,2}	A	-	35.86/10.22/29.69	32.06/9.04/25.16	14.66/2.26/11.44
Discourse-aware ²	A	14.3M	38.93/15.37/35.21	35.80/11.05/31.80	-/-
TLM ¹	A	234M	37.06/11.69/34.27	39.65/12.15/35.76	20.40/6.90/17.10
TLM ¹	H	234M	42.13/16.27/39.21	41.62/14.69/38.03	20.10/6.50/16.60
Mem2Mem (ours)	A	14.7M	42.06/ 16.56 /38.11	41.81/14.99 /37.24	20.00/6.60/16.80

would correlate with it. The initial state of the decoder memory \mathbf{M}_E is used to compute $\mathbf{c}_t^{(m)}$ since the representation of \mathbf{M}_D deviates from the original representation space of $\mathbf{h}_j^{(s)}$ due to the write operation. The final training objective of Mem2Mem becomes as follows.

$$\mathcal{L} + \lambda_1 \mathcal{L}^{(comp)} + \lambda_2 \mathcal{L}^{(read)} \quad (12)$$

where the weights for regularization λ_1 and λ_2 are hyperparameters.

4 RELATED WORK

Recent works in abstractive summarization have leveraged intermediate content selection. In these approaches, writing a summary is factorized into two steps: *extraction* and *generation*. An extractor is used to prioritize and select the most important part of the input text. The extractor is normally trained on a sequence of binary labels where each label indicates whether the corresponding text unit should be selected or not. The level of extraction can be word-level (Gehrmann et al., 2018; Cho et al., 2019) or sentence-level (Chen & Bansal, 2018; Hsu et al., 2018; Bae et al., 2019; Subramanian et al., 2019). As the ground truth for extraction is typically missing, heuristics that measure n-gram overlap with the target summary are used to build extractive oracles. Similar to other approaches, Mem2Mem performs sentence-level extraction to deal with long source articles². Mem2Mem determines the alignment between source and target sentences in a latent space without relying on possibly suboptimal extractive heuristics. In addition, sentence extraction is not sequentially done in Mem2Mem, which addresses the exposure bias issue (Ranzato et al., 2016).

Memory Augmented Encoder Decoder (MAED) architectures (Wang et al., 2016; Yogatama et al., 2018; Ma & Principe, 2019; Le et al., 2018) have been proposed for conditional natural language generation tasks, such as machine translation (Kaiser et al., 2017) and image captioning (Park et al., 2017). Using differentiable read and write operations to an external module, MAED can represent non-local context of RNNs with enhanced memory capacity. Such models are able to store temporally distant information of large input sequences, a feature that is particularly useful for long text summarization. In the context of short text abstractive summarization, Kim et al. (2019) proposed a memory architecture named multi-level memory networks (MMN). MMN can flexibly reduce representations at different levels of document hierarchy into a fixed size external memory. Authors used multi-layer dilated Convolutional Neural Networks (CNN) (Yu & Koltun, 2016; van den Oord et al., 2016) to build a hierarchical representation of the document. Mem2Mem also constructs

²In preliminary experiments, we applied word-level selection on the PubMed and arXiv datasets, which led to poor results

Table 2: Model ablation study on the PubMed dataset.

Model	ROUGE		
	1	2	L
Baseline HRED	40.02	15.82	36.28
+ Encoder Mem	40.62	15.89	36.85
+ Decoder Mem	40.51	15.9	36.64
+ Mem Transfer	41.27	16.24	37.38
+ Reg $\mathcal{L}^{(comp)}$	41.82	16.51	37.76
+ Reg $\mathcal{L}^{(read)}$	42.06	16.56	38.11

Table 3: ROUGE scores of unsupervised extractive methods on the PubMed and the arXiv dataset. The result of baseline extractive methods is from Subramanian et al. (2019).

Data	Model	ROUGE		
		1	2	L
PubMed	Lead-10	37.45	14.19	34.07
	LexRank	39.19	13.89	34.59
	Mem2Mem	41.77	16.20	37.81
	Gold Ext	47.76	20.36	39.19
arXiv	Lead-10	35.52	10.33	31.44
	LexRank	33.85	10.73	28.99
	Mem2Mem	41.76	14.95	37.22
	Gold Ext	44.25	18.17	35.33

memory from the hierarchical representation of the document, but by compressing it into a sparse set of sentence representations. Further, MMN’s memory representations remain static throughout the decoding process while Mem2Mem dynamically updates its memory, which is more effective in learning long term dependency. Lastly but not least, our work proposes novel regularization for memory read and compression.

5 RESULTS AND DISCUSSION

5.1 EXPERIMENT SETUP

We evaluate Mem2Mem on the PubMed, arXiv (Cohan et al., 2018) and Newsroom abstractive datasets (Grusky et al., 2018) which are large scale summarization datasets. The average lengths of source articles and target summaries are 3016/203 (PubMed), 4938/220 (arXiv), and 751/30 (Newsroom) respectively. They are up to 6 times longer than the widely used CNN/DailyMail dataset (781/56) (Hermann et al., 2015; See et al., 2017). Our pre-processing and training setups are identical to Cohan et al. (2018) and Subramanian et al. (2019). More details on training and evaluation can be found in Appendix 2. For quantitative evaluation, we use the ROUGE metric (Lin, 2004) and report F-1 ROUGE scores.

5.2 RESULTS

Table 1 shows the ROUGE scores on three summarization datasets. The hybrid type (H) refers to models that use two-step extractive-abstractive summarization. On the PubMed dataset, the TLM model shows the highest scores in R-1 and R-L. Mem2Mem is close to those scores and shows higher R-2 (+0.29) scores with 16 times less parameters (14.7M vs. 234M). It achieves such performance over the TLM model without ground truth labels for sentence extraction. We also reiterate that Mem2Mem is trained completely end-to-end whereas the hybrid TLM requires separate training for the extractor and the conditional transformer language model. We find similar results on the arXiv and Newsroom datasets. On the arXiv dataset, Mem2Mem even surpasses the transformer based TLM model in R-1 (+0.19) and R-2 (+0.3) scores. Mem2Mem also shows competitive results on the Newsroom abstractive dataset.

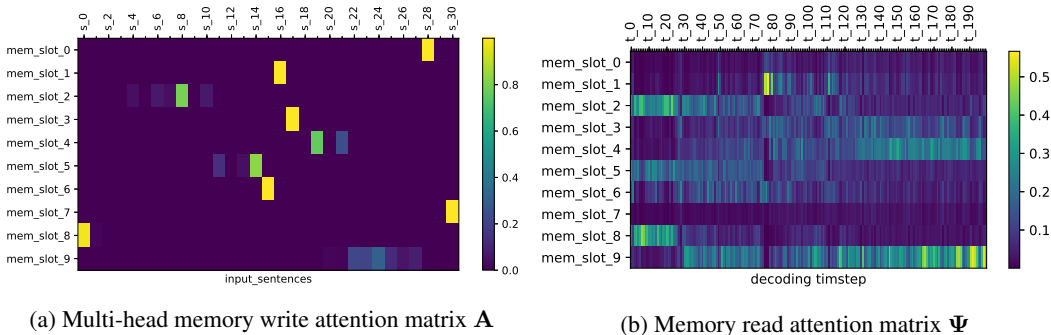


Figure 2: Multi-head memory write attention matrix \mathbf{A} and memory read attention matrix Ψ . In (a), rows denote memory heads or slots and columns indicate input sentence indices. In (b), columns indicate decoding time steps.

5.3 ABLATION STUDY

To assess the importance of Mem2Mem components, we conduct an ablation study on the PubMed dataset. Table 2 shows the effects of adding different Mem2Mem add-ons on ROUGE scores. *+Encoder Mem* is the baseline HRED augmented with the encoder memory described in section 3.1. The result demonstrates that memory context indeed enhances the performance measures of the generated summaries. *+Decoder Mem* adds the write operation to the memory but without memory transfer. In this case, the decoder memory \mathbf{M}_D is initialized with zeros not with the encoder memory \mathbf{M}_E . Compared to the Baseline HRED, the result shows that the write mechanism on the decoder memory helps the generation even without the memory transfer. This indicates that the summary writing process largely benefits from accessing long term contextual information of the output text.

Transferring memory (*+Mem Transfer*) brings substantial improvements in ROUGE scores. It seems to be crucial to initiate the summary generation from the selected memory representations, showing the importance of the memory compression and transfer steps. Furthermore, it can be observed that discouraging redundancies over the encoder memory head via $\mathcal{L}^{(comp)}$ leads to additional improvements on all ROUGE scores. Finally, adding another regularization $\mathcal{L}^{(read)}$ on the decoder memory read operation completes the Mem2Mem architecture and achieves the best ROUGE scores.

5.4 IMPLICIT EXTRACTION VIA MEMORY COMPRESSION

Our initial hypothesis was that the encoder memory \mathbf{M}_E would pick a set of the most salient input sentences for summarization. To confirm, we analyze the quality of the extractive summarization by memory compression. Concretely, we concatenate the sentences with the highest attention weight of each memory head to generate a summary. Table 3 shows the ROUGE scores of different unsupervised extractive summarization methods on the PubMed and arXiv datasets. The extractive summarization performed by the Mem2Mem’s memory compression outperforms existing unsupervised extractive summarization baselines. The result indicates that Mem2Mem’s memory compression is able to prioritize amongst a large set of input sentences without ground truth sentence labels.

5.5 DYNAMIC MEMORY READ BY DECODER

The benefit of implicit extraction is maximized when the extracted representations are properly consumed in the text generation. To understand the link between the representations stored in the memory and the summary generation, we analyze the memory read attention weights ψ_{tk} in equation (7) throughout the decoding. Figure 2 (b) shows that the Mem2Mem decoder fully utilizes all memory representations. We also find a pattern that memory read attention weights are mostly concentrated on the front part of the source text in the beginning of the decoding (mem slots 2,5,8), and gradually moves to the latter part (mem slots 3,4,9). This demonstrates Mem2Mem’s ability to update the read operation to dynamically capture relevant input contexts during the summary generation.

Table 4: Percentage ratios of output summary n-grams found in the PubMed original input article.

Models	n-grams			
	5	10	15	20
Reference	12.4	3.2	1.3	0.9
Baseline HRED	38.1	26.0	17.4	13.4
TLM-hybrid	24.7	10.8	7.0	4.1
Mem2Mem (ours)	28.3	19.2	17.6	10.7

Table 5: Results of human evaluation. Maximum score for each criterion is 5.

Models	Human evaluation scores			
	COH	INF	REL	FLU
Baseline HRED	3.02	2.98	2.85	3.08
TLM-hybrid	3.67	3.11	3.43	3.51
Mem2Mem (ours)	3.59	3.13	3.46	3.38

5.6 ABSTRACTIVENESS OF THE SUMMARY

To analyze the abstractiveness of generated summaries, we present the ratio of output summary n-grams present in the original input article. Table 4 shows that Mem2Mem copies less n-grams than the baseline HRED. Compared to the baseline HRED, Mem2Mem generates approximately 10% more novel 5-grams and 7% more 10-grams respectively. The result along with higher ROUGE scores highlights the Mem2Mem’s ability to generate novel words for abstractive summarization while staying focused on important parts of the article. Although TLM (Subramanian et al., 2019)³ shows the highest abstractiveness, Mem2Mem achieves its result with a significantly smaller model.

5.7 HUMAN EVALUATION

We also perform a human evaluation to assess the quality of generated summaries. For the human evaluation, 20 random arXiv testset article-summary pairs are presented to three Amazon Mechanical Turk workers. Workers judge generated summaries on four different aspects: Coherence (COH: does the summary make sense), Informativeness (INF: are the most important points of the article captured), Redundancy (RED: does the summary repeat itself), and Fluency (FLU: how fluent is the summary). Table 5 shows that Mem2Mem obtains the highest scores in Informativeness and Redundancy. Its Coherence score is also close to the best score by TLM. Fluency is an advantage for the transformer-based TLM as expected but Mem2Mem is close and still greatly superior to vanilla hierarchical encoder decoders. While the ROUGE difference between the baseline and Mem2Mem are substantial in Table 2, the results of the human evaluation show more pronounced differences in the quality of generated text. For output summary examples, please refer to Table 6 in Appendix 4.

6 CONCLUSION

This work proposes Mem2Mem, a novel MAED based mechanism for very long text abstractive summarization. Mem2Mem involves two memory types: A static encoder memory for compressing input texts and a dynamic decoder memory which refines the generation process. Memory transfer between them links two memories and maximizes the benefit of content extraction aimed for summarization. Different from existing hybrid extractive and abstractive approaches, Mem2Mem incorporates an extraction step without ground truth sentence labels and multi-step training. We demonstrate the effectiveness of Mem2Mem by showing promising results on the PubMed, arXiv, and Newsroom summarization datasets with an order of magnitude less parameters than competing transformer-based models. The Mem2Mem’s memory compression can be generalized to other domains that require text generation guided by content selection. In future work, we will extend and validate the strength of our approach on a variety of language learning tasks.

³The authors provided example summaries from their model.

REFERENCES

- Sanghwan Bae, Taeuk Kim, Jihoon Kim, and Sang-goo Lee. Summary level training of sentence rewriting for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 10–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5402. URL <https://www.aclweb.org/anthology/D19-5402>.
- Yen-Chun Chen and Mohit Bansal. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 675–686, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1063. URL <https://www.aclweb.org/anthology/P18-1063>.
- Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 484–494, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1046. URL <https://www.aclweb.org/anthology/P16-1046>.
- Jaemin Cho, Minjoon Seo, and Hannaneh Hajishirzi. Mixture content selection for diverse sequence generation. 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)*, pp. 3121–3131, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1308. URL <https://www.aclweb.org/anthology/D19-1308>.
- 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. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL <https://www.aclweb.org/anthology/D14-1179>.
- Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 615–621, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2097. URL <https://www.aclweb.org/anthology/N18-2097>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://www.aclweb.org/anthology/N19-1423>.
- Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4098–4109, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1443. URL <https://www.aclweb.org/anthology/D18-1443>.
- Max Grusky, Mor Naaman, and Yoav Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 708–719, 2018.
- Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1631–1640, Berlin, Germany, August

2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1154. URL <https://www.aclweb.org/anthology/P16-1154>.
- 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 28*, pp. 1693–1701. 2015. URL <http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.pdf>.
- Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 132–141, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1013. URL <https://www.aclweb.org/anthology/P18-1013>.
- Lukasz Kaiser, Ofir Nachum, Aurko Roy, and Samy Bengio. Learning to remember rare events. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL <https://openreview.net/forum?id=SJTQLdqlg>.
- Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. Abstractive summarization of Reddit posts with multi-level memory networks. 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)*, pp. 2519–2531, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1260. URL <https://www.aclweb.org/anthology/N19-1260>.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1412.6980>.
- Hung Le, Truyen Tran, Thin Nguyen, and Svetha Venkatesh. Variational memory encoder-decoder. In *Advances in Neural Information Processing Systems 31*, pp. 1508–1518. 2018. URL <http://papers.nips.cc/paper/7424-variational-memory-encoder-decoder.pdf>.
- C. Y. Lin. Looking for a few good metrics: Automatic summarization evaluation - how many samples are enough? In *Proceedings of the NTCIR Workshop 4*, 2004.
- Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL <https://www.aclweb.org/anthology/D15-1166>.
- Y. Ma and J. C. Principe. A taxonomy for neural memory networks. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2019.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Guçlçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290, Berlin, Germany, August 2016a. Association for Computational Linguistics. doi: 10.18653/v1/K16-1028. URL <https://www.aclweb.org/anthology/K16-1028>.
- Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. Classify or select: Neural architectures for extractive document summarization. *arXiv preprint arXiv:1611.04244*, 2016b.
- Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1747–1759, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1158. URL <https://www.aclweb.org/anthology/N18-1158>.

- Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. Attend to you: Personalized image captioning with context sequence memory networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6432–6440, 2017.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL <http://arxiv.org/abs/1511.06732>.
- Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL <https://www.aclweb.org/anthology/P17-1099>.
- Sandeep Subramanian, Raymond Li, Jonathan Pilault, and Christopher Pal. On extractive and abstractive neural document summarization with transformer language models. *arXiv preprint arXiv:1909.03186*, 2019.
- Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Coverage-based neural machine translation. *CoRR*, abs/1601.04811, 2016. URL <http://arxiv.org/abs/1601.04811>.
- Aáron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016. URL <https://arxiv.org/abs/1609.03499>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017. URL <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems 28*, pp. 2692–2700. 2015. URL <http://papers.nips.cc/paper/5866-pointer-networks.pdf>.
- Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. Memory-enhanced decoder for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 278–286, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1027. URL <https://www.aclweb.org/anthology/D16-1027>.
- Dani Yogatama, Yishu Miao, Gábor Melis, Wang Ling, Adhiguna Kuncoro, Chris Dyer, and Phil Blunsom. Memory architectures in recurrent neural network language models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL <https://openreview.net/forum?id=SkFqf01AZ>.
- Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL <http://arxiv.org/abs/1511.07122>.

A APPENDIX

A.1 BASELINE ARCHITECTURE

In addition to the hierarchical recurrent encoder-decoder (HRED) architecture described in section 2, the following features are used for both baseline and Mem2Mem architectures.

A.1.1 POINTER GENERATOR NETWORK

In order to handle out-of-vocabulary (OOV) token predictions, pointer generator in See et al. (2017) is used to copy words directly from the input document. At each step t , the decoder decides whether to predict the next target word from the input text or the generation mechanism. The pointer generator computes z_t which denotes the probability of choosing P_g for sampling the next target token.

$$z_t = \text{softmax}(w_c^T \mathbf{c}_t + w_d^T \mathbf{h}_t^{(d)} + w_x^T \mathbf{x}'_t) \quad (13)$$

where \mathbf{x}'_t is the embedding of the previous target token. The probability z_t is used as a variable for soft switch between generating a word from the vocabulary (P_g) or directly copying from the source document (P_c). The probability of copying a word w from the source text is calculated based on the attention weights γ .

$$P_c(y_t = w | y_{1:t-1}) = \sum_{i:x_i=w} \gamma_{ti} \quad (14)$$

Note that $P_c(y_t = w | y_{1:t-1}) = 0$ if w does not exist in the source document. Likewise, $P_g(y_t = w | y_{1:t-1}) = 0$ if w is an out of vocabulary word. Combining two probability distributions, the final probability of the next word y_t being w is as follows.

$$P(y_t = w | y_{1:t-1}) = z_t P_g(y_t = w | y_{1:t-1}) + (1 - z_t) P_c(y_t = w | y_{1:t-1}) \quad (15)$$

A.1.2 DECODER COVERAGE

It is well known that RNN sequence-to-sequence models tend to suffer from repeated phrases when generating long target sequences. See et al. (2017) tackled this issue by keeping track of the attention coverage. More concretely, the coverage vector \mathbf{cov}_t at the decoding step t is computed by taking the summation of the token-level attention weights α until the last step $t - 1$.

$$\mathbf{cov}_t = \sum_{t'=0}^{t-1} \alpha_{t'} \quad (16)$$

To inform the decoder of the history of attention weights, the coverage vector is fed into the token-level attention mechanism, which modifies the equation (3) to the following equation.

$$\alpha_{ti} = \text{softmax}\left(v^T \tanh\left(W_e \mathbf{h}_i^{(w)} + W_d \mathbf{h}_t^{(d)} + w_c \mathbf{cov}_t^T\right)\right) \quad (17)$$

A.2 TRAINING DETAILS

We generally follow the pre-processing steps in Cohan et al. (2018) for the PubMed and arXiv datasets. The maximum number of sections is set to 4 and the maximum number of tokens for each section is 500. The length of the target summary is limited to 200 tokens.

Single-layer bidirectional GRUs (Cho et al., 2014) are used for the sentence and the document encoders. The decoder is also a single layer GRU. All GRUs have the hidden size of 256. The dimensionality of token embeddings is 128 and embeddings are trained from scratch. The vocabulary size is limited to 50,000. Batch size is 16 and Adam (Kingma & Ba, 2015) with learning rate $2e^{-4}$ is used for training. Maximum gradient norm is set to 2. We train all models for 15 epochs. At the test time, beam search with the beam size 4 is used for decoding.

For Mem2Mem hyperparameters, the number of heads for the memory compression is 10 and the self-attention hidden size is 128. The weights λ_1 and λ_2 for the regularization $\mathcal{L}^{(comp)}$ and $\mathcal{L}^{(read)}$ are 0.0001 and 0.01 respectively.

A.3 THE EFFECT OF REGULARIZATION

The following figures show the effect of regularization in Mem2Mem.

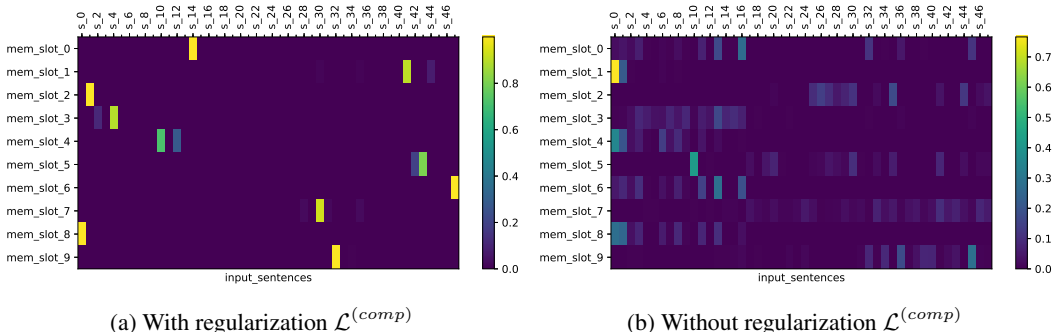


Figure 3: The effect of regularization on memory compression. Examples of the multi-head encoder memory write attention matrix \mathbf{A} are illustrated. Rows denote memory heads or slots and columns indicate input sentence indices. Note that the regularization loss $\mathcal{L}^{(comp)}$ removes the redundancy over different memory heads and guides each slot to focus on a single sentence.

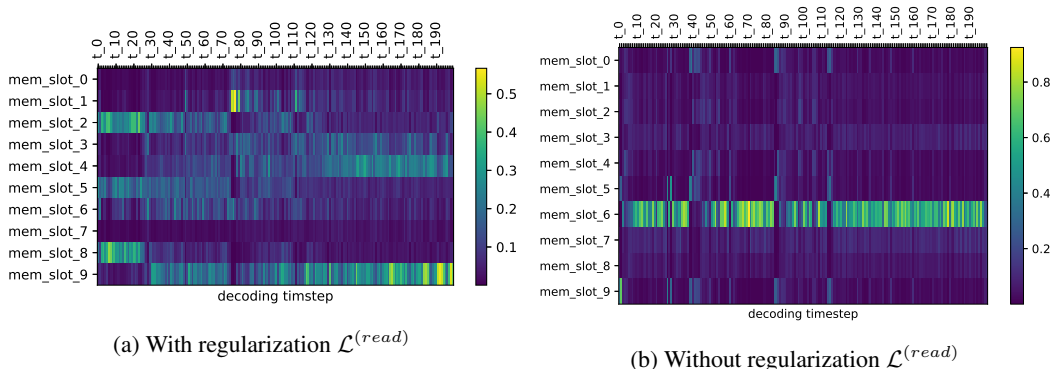


Figure 4: The effect of regularization on memory read. Examples of the decoder memory read attention matrix $\mathbf{\Psi}$ are illustrated. Rows denote memory heads or slots and columns indicate decoding time steps. Note that the regularization loss $\mathcal{L}^{(read)}$ encourages the model to fully utilize the compressed memory representations.

A.4 QUALITATIVE RESULTS

Reference and system generated summaries from the test set of the arXiv dataset are shown in the following table. Compared to the baseline, Mem2Mem shows much less repetition and produces more concise summaries. Even the TLM model repeats several phrases or sentences.

Table 6: Reference and system generated summaries from the test set of the arXiv dataset.

<p>Reference — we study the behavior of simple principal pivoting methods for the p-matrix linear complementarity problem (p-lcp). we solve an open problem of morris by showing that murty’s least index pivot rule (under any fixed index order) leads to a quadratic number of iterations on morris’s highly cyclic p-lcp examples. we then show that on k-matrix lcp instances, all pivot rules require only a linear number of iterations. as the main tool, we employ unique sink orientations of cubes, a useful combinatorial abstraction of the p-lcp.</p>
<p>Baseline HRED — the third author of this paper still vividly recollects his visit to victor klee at the university of washington (seattle) in august 2000. the third author of this paper still vividly recollects his visit to victor klee at the university of washington (seattle) in august 2000. the third author of this paper still vividly recollects his visit to victor klee at the university of washington (seattle) in august 2000. in this paper, we introduce the digraph model behind p-lcps and show that the simplex method is polynomial-time for which the number of iterations is exponential in the number of variables and constraints.</p>
<p>TLM-hybrid — we study linear complementarity problems with sufficient matrices and the criss-cross method. we use a novel notion of linear complementarity to derive linear complementarity problems with sufficient matrices and the “linear” version of the “quadratic” version of the “linear” version of the “linear” version. this yields a new family of np-complete problems for which both linear complementarity and the “quadratic” version of the “linear” version of the “quadratic” version are possible.</p>
<p>Mem2Mem — in this paper we introduce a digraph model behind p-lcps and show that the simplex method can be used to solve a linear program (lp) for which the number of iterations is exponential in the number of variables and constraints on the number of iterations. we show that the number of arithmetic operations does not depend on the bit lengths of the input numbers. we also show that the number of iterations required to solve the linear complementarity problem can be used to determine the expected performance of the lp.</p>
<p>Reference — we have developed a procedure for the classification of eclipsing binaries from their light-curve parameters and spectral type. the procedure was tested on more than 1000 systems with known classification, and its efficiency was estimated for every evolutionary status we use. the procedure was applied to about 4700 binaries with no classification, and the vast majority of them was classified successfully. systems of relatively rare evolutionary classes were detected in that process, as well as systems with unusual and/or contradictory parameters. also, for 50 previously unclassified cluster binaries evolutionary classes were identified. these stars can serve as tracers for age and distance estimation of their parent stellar systems. the procedure proved itself as fast, flexible and effective enough to be applied to large ground based and space born surveys, containing tens of thousands of eclipsing binaries.</p>
<p>Baseline HRED — in this paper we present a novel procedure for determination of the evolutionary status of eclipsing binaries. the procedure is based on the most comprehensive set of rules for the classification of eclipsing binaries, which is the world’s principal database of eclipsing binary systems with available classification. the main goal of this paper is to develop a fast and effective procedure for determination of the binary’s spectral type or color index. the procedure is based on the most comprehensive set of rules for the classification of eclipsing binaries, which is the world’s principal database of eclipsing binary systems with available classification. the main goal of this paper is to develop a fast and effective procedure for determination of the evolutionary status of eclipsing binaries. the method is based on a catalogue which includes a number and the eclipsing variables, and a classification procedure. we find a good fit of all known parameters, which can not have any value in a large range. we find a significant fraction (mp, depth) for each of these systems, and that a given number is @xmath0 mag</p>
<p>TLM-hybrid — a fast and effective procedure for determination of the evolutionary status of eclipsing binaries is presented. a complete set of rules is developed that utilizes all available data for a specific class of eclipsing binaries, while requiring only light curve parameters and an estimate of the binary’s spectral type or color index. the procedure is tested with the catalogue of eclipsing binaries (cev), which is the world’s principal database of eclipsing binaries with available classification. the main goal of our work is to develop a fast and effective procedure for determination of the evolutionary status of eclipsing binaries. we apply it to cev binaries with already available classification. the main objective of this work is to develop a fast and effective procedure for determination of the evolutionary status of eclipsing binaries. the main objective of our work is to develop a fast and effective procedure for determination of the evolutionary status of eclipsing binaries. we apply it to cev binaries with already available classification. the main objective of this work is to demonstrate the usefulness of our method, and to propose a possible application of it to cev binaries with already known classification.</p>
<p>Mem2Mem — in this paper we present a novel procedure for determination of the evolutionary status of eclipsing binaries, while requiring only light curve parameters and an estimate of the binary’s spectral type or color index. the procedure is based on the most comprehensive set of rules for the classification of eclipsing binaries, which can be used to quickly characterize large numbers of eclipsing binaries. the procedure is tested with the catalogue of eclipsing variables (dr & sh), which is the world’s principal database of eclipsing binary systems with available classification. we find that the number of fully characterized eclipsing binaries can be used to quickly characterize large numbers of eclipsing binaries.</p>