Multi-document Summarization with Maximal Marginal Relevance-guided Reinforcement Learning

Yuning Mao[1], **Yanru Qu**[1], **Yiqing Xie**[1], **Xiang Ren**[2], **Jiawei Han**[1]
[1]University of Illinois at Urbana-Champaign, IL, USA
[2]University of Southern California, CA, USA
[1]{yuningm2, yanruqu2, xyiqing2, hanj}@illinois.edu [2]xiangren@usc.edu

Abstract

While neural sequence learning methods have made significant progress in single-document summarization (SDS), they produce unsatisfactory results on multi-document summarization (MDS). We observe two major challenges when adapting SDS advances to MDS: (1) MDS involves larger search space and yet more limited training data, setting obstacles for neural methods to learn adequate representations; (2) MDS needs to resolve higher information redundancy among the source documents, which SDS methods are less effective to handle. To close the gap, we present RL-MMR, Maximal Margin Relevance-guided Reinforcement Learning for MDS, which unifies advanced neural SDS methods and statistical measures used in classical MDS. RL-MMR casts MMR guidance on fewer promising candidates, which restrains the search space and thus leads to better representation learning. Additionally, the explicit redundancy measure in MMR helps the neural representation of the summary to better capture redundancy. Extensive experiments demonstrate that RL-MMR achieves state-of-the-art performance on benchmark MDS datasets. In particular, we show the benefits of incorporating MMR into end-to-end learning when adapting SDS to MDS in terms of both learning effectiveness and efficiency.[1]

1 Introduction

Text summarization aims to produce condensed summaries covering salient and non-redundant information in the source documents. Recent studies on single-document summarization (SDS) benefit from the advances in neural sequence learning (Nallapati et al., 2016; See et al., 2017; Chen and Bansal, 2018; Narayan et al., 2018) as well as pre-trained language models (Liu and Lapata, 2019; Lewis et al., 2019; Zhang et al., 2020) and make great progress. However, in multi-document summarization (MDS) tasks, neural models are still facing challenges and often underperform classical statistical methods built upon handcrafted features (Kulesza and Taskar, 2012).

We observe two major challenges when adapting advanced neural SDS methods to MDS: (1) **Large search space**. MDS aims at producing summaries from multiple source documents, which exceeds the capacity of neural SDS models (See et al., 2017) and sets learning obstacles for adequate representations, especially considering that MDS labeled data is more limited. For example, there are 287K training samples (687 words on average) on the CNN/Daily Mail SDS dataset (Nallapati et al., 2016) and only 30 on the DUC 2003 MDS dataset (6,831 words). (2) **High redundancy**. In MDS, the same statement or even sentence can spread across different documents. Although SDS models adopt attention mechanisms as implicit measures to reduce redundancy (Chen and Bansal, 2018), they fail to handle the much higher redundancy of MDS effectively (Sec. 4.2.3).

There have been attempts to solve the aforementioned challenges in MDS. Regarding the **large search space**, prior studies (Lebanoff et al., 2018; Zhang et al., 2018) perform sentence filtering using a sentence ranker and only take top-ranked K sentences. However, such a hard cutoff of the search space makes these approaches insufficient in the exploration of the (already scarce) labeled data and limited by the ranker since most sentences are discarded,[2] albeit the discarded sentences are important and could have been favored. As a result, although these studies perform better than directly applying their base SDS models (See et al.,

[1]Code can be found at https://github.com/morningmoni/RL-MMR.

[2]K is set to 7 in Lebanoff et al. (2018) and 15 in Zhang et al. (2018). One document set in DUC 2004 (Paul and James, 2004), for example, averages 265.4 sentences.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1737–1751,

2017; Tan et al., 2017) to MDS, they do not outperform state-of-the-art MDS methods (Gillick and Favre, 2009; Kulesza and Taskar, 2012). Regarding the **high redundancy**, various redundancy measures have been proposed, including heuristic post-processing such as counting new bi-grams (Cao et al., 2016) and cosine similarity (Hong et al., 2014), or dynamic scoring that compares each source sentence with the current summary like Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998). Nevertheless, these methods still use lexical features without semantic representation learning. One extension (Cho et al., 2019) of these studies uses capsule networks (Hinton et al., 2018) to improve redundancy measures. However, its capsule networks are pre-trained on SDS and fixed as feature inputs of classical methods without end-to-end representation learning.

In this paper, we present a deep RL framework, MMR-guided Reinforcement Learning (RL-MMR) for MDS, which unifies advances in SDS and one classical MDS approach, MMR (Carbonell and Goldstein, 1998) through end-to-end learning. RL-MMR addresses the MDS challenges as follows: (1) RL-MMR overcomes the **large search space** through *soft attention*. Compared to hard cutoff, our soft attention favors top-ranked candidates of the sentence ranker (MMR). However, it does not discard low-ranked ones, as the ranker is imperfect, and those sentences ranked low may also contribute to a high-quality summary. Soft attention restrains the search space while allowing more exploration of the limited labeled data, leading to better representation learning. Specifically, RL-MMR infuses the entire prediction of MMR into its neural module by attending (restraining) to important sentences and downplaying the rest instead of completely discarding them. (2) RL-MMR resolves the **high redundancy** of MDS in a unified way: the explicit redundancy measure in MMR is incorporated into the neural representation of the current state, and the two modules are coordinated by RL reward optimization, which encourages non-redundant summaries.

We conduct extensive experiments and ablation studies to examine the effectiveness of RL-MMR. Experimental results show that RL-MMR achieves state-of-the-art performance on the DUC 2004 (Paul and James, 2004) and TAC 2011 (Owczarzak and Dang, 2011) datasets (Sec. 4.2.1). A comparison between various combination mechanisms demonstrates the benefits of soft attention in the large search space of MDS (Sec. 4.2.2). In addition, ablation and manual studies confirm that RL-MMR is superior to applying either RL or MMR to MDS alone, and MMR guidance is effective for redundancy avoidance (Sec. 4.2.3).

Contributions. (1) We present an RL-based MDS framework that combines the advances of classical MDS and neural SDS methods via end-to-end learning. (2) We show that our proposed soft attention is better than the hard cutoff of previous methods for learning adequate neural representations. Also, infusing the neural representation of the current summary with explicit MMR measures significantly reduces summary redundancy. (3) We demonstrate that RL-MMR achieves new state-of-the-art results on benchmark MDS datasets.

2 Problem Formulation

We define $\mathcal{D} = \{D_1, D_2, ..., D_N\}$ as a set of documents on the same topic. Each document set $\mathcal{D}$ is paired with a set of (human-written) reference summaries $\mathcal{R}$. For the convenience of notation, we denote the j-th sentence in $\mathcal{D}$ as s_j when concatenating the documents in $\mathcal{D}$. We focus on extractive summarization where a subset of sentences in $\mathcal{D}$ are extracted as the system summary $\boldsymbol{E}$. A desired system summary $\boldsymbol{E}$ covers salient and non-redundant information in $\mathcal{D}$. $\boldsymbol{E}$ is compared with the reference summaries $\mathcal{R}$ for evaluation.

3 The RL-MMR Framework

Overview. At a high level, RL-MMR infuses MMR guidance into end-to-end training of the neural summarization model. RL-MMR uses hierarchical encoding to efficiently encode the sentences in multiple documents and obtains the neural sentence representation $\boldsymbol{A}_j$. RL-MMR models salience by combining MMR and sentence representation $\boldsymbol{A}_j$, and measures redundancy by infusing MMR with neural summary representation $\boldsymbol{z}_t$, which together form the state representation $\boldsymbol{g}_t$. At each time step, one sentence is extracted based on the MMR-guided sentence representation and state representation, and compared with the reference, the result (reward) of which is then back-propagated for the learning of both neural representation and MMR guidance. An illustrative figure of RL-MMR is shown in Fig. 1.

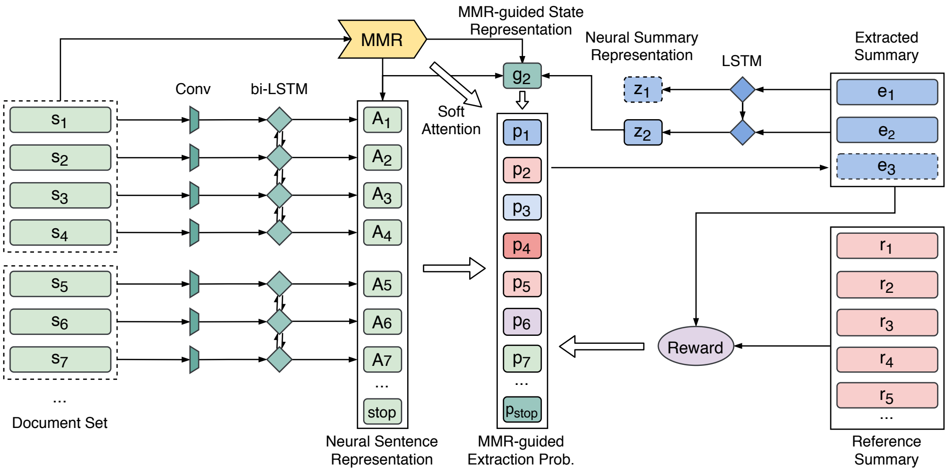

Figure 1: **An overview of the proposed MDS framework RL-MMR**. Neural sentence representation $\boldsymbol{A}_j$ is obtained through sentence-level convolutional encoder and document-level bi-LSTM encoder. MMR guidance is incorporated into neural sentence representation $\boldsymbol{A}_j$ and state representation $\boldsymbol{g}_t$ through soft attention and end-to-end learned through reward optimization.

In the following, we first describe MMR and the neural sentence representation. We then introduce the neural sentence extraction module and how we incorporate MMR guidance into it for better MDS performance. Finally, we illustrate how neural representation and MMR guidance are jointly learned via end-to-end reinforcement learning.

3.1 Maximal Marginal Relevance

Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998) is a general summarization framework that balances summary salience and redundancy. Formally, MMR defines the score of a sentence s_j at time t as $m_j^t = \lambda \mathbf{S}(s_j, \mathcal{D}) - (1-\lambda)\max_{e \in \boldsymbol{E}_t} \mathbf{R}(s_j, e)$, where $\lambda \in [0, 1]$ is the weight balancing salience and redundancy. $\mathbf{S}(s_j, \mathcal{D})$ measures how salient a sentence s_j is, estimated by the similarity between s_j and $\mathcal{D}$. $\mathbf{S}(s_j, \mathcal{D})$ does not change during the extraction process. $\boldsymbol{E}_t$ consists of sentences that are already extracted before time t. $\max_{e \in \boldsymbol{E}_t} \mathbf{R}(s_j, e)$ measures the redundancy between s_j and each extracted sentence e and finds the most redundant pair. $\max_{e \in \boldsymbol{E}_t} \mathbf{R}(s_j, e)$ is updated as the size of $\boldsymbol{E}_t$ increases. Intuitively, if s_j is similar to any sentence $e \in \boldsymbol{E}_t$, it would be deemed redundant and less favored by MMR. There are various options regarding the choices of $\mathbf{S}(s_j, \mathcal{D})$ and $\mathbf{R}(s_j, e)$, which we compare in Sec. 4.2.4.

We denote the (index of the) sentence extracted at time t as e_t. MMR greedily extracts one sentence at a time according to the MMR score: $e_t = \arg\max_{s_j \in \mathcal{D} \setminus \boldsymbol{E}_t} m_j^t$. Heuristic and deterministic algorithms like MMR are rather efficient and work reasonably well in some cases. However, they lack holistic modeling of summary quality and the capability of end-to-end representation learning.

3.2 Neural Sentence Representation

To embody end-to-end representation learning, we leverage the advances in SDS neural sequence learning methods. Unlike prior studies on adapting SDS to MDS (Lebanoff et al., 2018), which concatenates all the documents chronologically and encodes them sequentially, we adapt hierarchical encoding for better efficiency and scalability. Specifically, we first encode each sentence s_j via a CNN (Kim, 2014) to obtain its sentence representation. We then separately feed the sentence representations in *each document* D_i to a bi-LSTM (Huang et al., 2015). The bi-LSTM generates a contextualized representation for each sentence s_j, denoted by $\boldsymbol{h}_j$. We form an action matrix $\boldsymbol{A}$ using $\boldsymbol{h}_j$, where the j-th row $\boldsymbol{A}_j$ corresponds to the j-th sentence (s_j) in $\mathcal{D}$. A pseudo sentence indicating the STOP action, whose representation is randomly initialized, is also included in $\boldsymbol{A}$, and sentence extraction is finalized when the STOP action is taken (Mao et al., 2018, 2019).

3.3 Neural Sentence Extraction

We briefly describe the SDS sentence extraction module (Chen and Bansal, 2018) that we base our work on, and elaborate in Sec 3.4 how we adapt it for better MDS performance with MMR guidance.

The probability of neural sentence extraction is measured through a two-hop attention mechanism. Specifically, we first obtain the neural summary representation $\boldsymbol{z}_t$ by feeding previously extracted sentences ($\boldsymbol{A}_{e_i}$) to an LSTM encoder. A time-dependent state representation $\boldsymbol{g}_t$ that considers both sentence representation $\boldsymbol{A}_j$ and summary representation $\boldsymbol{z}_t$ is obtained by the *glimpse* operation (Vinyals et al., 2016):

$$a_j^t = \boldsymbol{v}_1^\intercal \tanh(\boldsymbol{W}_1 \boldsymbol{A}_j + \boldsymbol{W}_2 \boldsymbol{z}_t), \tag{1}$$

$$\boldsymbol{\alpha}^t = \text{softmax}(\boldsymbol{a}^t), \tag{2}$$

$$\boldsymbol{g}_t = \sum_j \boldsymbol{\alpha}_j^t \boldsymbol{W}_1 \boldsymbol{A}_j, \tag{3}$$

where $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{v}_1$ are model parameters. $\boldsymbol{a}^t$ represents the vector composed of a_j^t. With $\boldsymbol{z}_t$, the attention weights $\boldsymbol{\alpha}_j^t$ are aware of previous extraction. Finally, the sentence representation $\boldsymbol{A}_j$ is attended again to estimate the extraction probability.

$$p_j^t = \begin{cases} \boldsymbol{v}_2^\intercal \tanh(\boldsymbol{W}_3 \boldsymbol{A}_j + \boldsymbol{W}_4 \boldsymbol{g}_t) & \text{if } s_j \neq e_i, \forall i < t \\ -\infty & \text{otherwise,} \end{cases} \tag{4}$$

where $\boldsymbol{W}_3$, $\boldsymbol{W}_4$, $\boldsymbol{v}_2$ are model parameters and previously extracted sentences $\{e_i\}$ are excluded.

The summary redundancy here is handled implicitly by $\boldsymbol{g}_t$. Supposedly, a redundant sentence s_j would receive a low attention weight a_j^t after comparing $\boldsymbol{A}_j$ and $\boldsymbol{z}_t$ in Eq. 1. However, we find such latent modeling insufficient for MDS due to its much higher degree of redundancy. For example, when news reports start with semantically similar sentences, using latent redundancy avoidance alone leads to repeated summaries (App B Table 8). Such observations motivate us to incorporate MMR, which models redundancy explicitly, to guide the learning of sentence extraction for MDS.

3.4 MMR-guided Sentence Extraction

In this section, we describe several strategies of incorporating MMR into sentence extraction, which keeps the neural representation for expressiveness while restraining the search space to fewer promising candidates for more adequate representation learning under limited training data.

Hard Cutoff. One straightforward way of incorporating MMR guidance is to only allow extraction from the top-ranked sentences of MMR. We denote the sentence list ranked by MMR scores m_j^t as $\boldsymbol{M}^t$. Given p_j^t – the neural probability of sentence extraction before softmax, we set the probability of the sentences after the first K sentences in $\boldsymbol{M}^t$ to $-\infty$. In this way, the low-ranked sentences in MMR are never selected and thus never included in the extracted summary. We denote this variant as RL-MMR$_{\text{HARD-CUT}}$.

There are two limitations of conducting hard cutoff in the hope of adequate representation learning: **(L1)** Hard cutoff ignores the values of MMR scores and simply uses them to make binary decisions. **(L2)** While hard cutoff reduces the search space, the decision of the RL agent is limited as it cannot extract low-ranked sentences and thus lacks exploration of the (already limited) training data. To tackle L1, a simple fix is to *combine* the MMR score m_j^t with the extraction probability measured by the neural sentence representation.

$$p_j^t = \begin{cases} \beta \boldsymbol{v}_2^\intercal \tanh(\boldsymbol{W}_3 \boldsymbol{A}_j + \boldsymbol{W}_4 \boldsymbol{g}_t) + (1-\beta)\text{FF}(m_j^t) \\ \qquad \text{if } s_j \neq e_i, \forall i < t \text{ and } s_j \in \boldsymbol{M}_{1:K}^t \\ -\infty \qquad \text{otherwise,} \end{cases} \tag{5}$$

where $\beta \in [0, 1]$ is a constant. FF$(\cdot)$ is a two-layer feed-forward network that enables more flexibility than using raw MMR scores, compensating for the magnitude difference between the two terms. We denote this variant as RL-MMR$_{\text{HARD-COMB}}$.

Soft Attention. To deal with L2, we explore soft variants that do not completely discard the low-ranked sentences but encourage the extraction of top-ranked sentences. The first variant, RL-MMR$_{\text{SOFT-COMB}}$, removes the constraint of $s_j \in \boldsymbol{M}_{1:K}^t$ in Eq. 5. This variant solves L2 but may re-expose the RL agent to L1 since its MMR module and neural module are loosely coupled and there is a learnable layer in their combination.

Therefore, we design a second variant, RL-MMR$_{\text{SOFT-ATTN}}$, which addresses both L1 and L2 by tightly incorporating MMR into neural representation learning via soft attention. Specifically, the MMR scores are first transformed and normalized: $\boldsymbol{\mu}^t = \text{softmax}(\text{FF}(\boldsymbol{m}^t))$, and then used to attend neural sentence representation $\boldsymbol{A}_j$ before the two-hop attention: $\boldsymbol{A}_j' = \boldsymbol{\mu}_j^t \boldsymbol{A}_j$. The state representation $\boldsymbol{g}_t$, which captures summary redundancy, is also impacted by MMR through the attention between summary representation $\boldsymbol{z}_t$ and MMR-guided sentence representation $\boldsymbol{A}_j'$ in Eq. 1. L1 is addressed as $\boldsymbol{\mu}^t$ represents the extraction probability estimated by MMR. L2 is resolved since the top-ranked sentences in MMR receive high attention, which empirically is enough to restrain the decision of the RL agent, while the low-ranked sentences are downplayed but not discarded, allowing more exploration of the search space.

3.5 MDS with Reinforcement Learning

The guidance of MMR is incorporated into neural representation learning through end-to-end RL training. Specifically, we formulate extractive MDS as a Markov Decision Process, where the state is defined by $(\mathcal{D} \setminus \boldsymbol{E}_t, \boldsymbol{g}_t)$. At each time step, one action is sampled from $\boldsymbol{A}$ given p_j^t, and its reward is measured by comparing the extracted sentence e_t with the reference $\mathcal{R}$ via ROUGE (Lin, 2004), *i.e.*, $r_t = \text{ROUGE-L}_{\text{F1}}(e_t, \mathcal{R})$. At the final step T when the STOP action is taken, an overall estimation of the summary quality is measured by $r_T = \text{ROUGE-1}_{\text{F1}}(\boldsymbol{E}, \mathcal{R})$. Reward optimization encourages salient and non-redundant summaries – intermediate rewards focus on the sentence salience of the current extracted sentence and the final reward captures the salience and redundancy of the entire summary.

Similar to prior RL-based models on SDS (Paulus et al., 2018; Chen and Bansal, 2018; Narayan et al., 2018), we use policy gradient (Williams, 1992) as the learning algorithm for model parameter updates. In addition, we adopt Advantage Actor-Critic (A2C) optimization – a critic network is added to enhance the stability of vanilla policy gradient. The critic network has a similar architecture to the one described in Sec. 3.2 and uses the sentence representation $\boldsymbol{A}$ to generate an estimation of the discounted reward, which is then used as the baseline subtracted from the actual discounted reward before policy gradient updates.

4 Experiments

We conduct extensive experiments to examine RL-MMR with several key questions: **(Q1)** How does RL-MMR perform compared to state-of-the-art methods? **(Q2)** What are the advantages of soft attention over hard cutoff in learning adequate neural representations under the large search space? **(Q3)** How crucial is the guidance of MMR for adapting SDS to MDS in the face of high redundancy?

4.1 Experimental Setup

Datasets. We take the MDS datasets from DUC and TAC competitions which are widely used in prior studies (Kulesza and Taskar, 2012; Lebanoff et al., 2018). Following convention (Wang et al., 2017; Cao et al., 2017; Cho et al., 2019), DUC 2004 (trained on DUC 2003) and TAC 2011 (trained on TAC 2008-2010) are used as the test sets. We use DUC 2004 as the validation set when evaluated on TAC 2011 and vice versa. More details of the dataset statistics are in App. A.1.

Evaluation Metrics. In line with recent work (Li et al., 2017; Lebanoff et al., 2018; Zhang et al., 2018; Cho et al., 2019), we measure ROUGE-1/2/SU4 F1 scores (Lin, 2004). The evaluation parameters are set according to Hong et al. (2014) with stemming and stopwords not removed. The output length is limited to 100 words. These setups are the same for all compared methods.[3]

Compared Methods. We compare RL-MMR with both classical and neural MDS methods. Note that *some previous methods are incomparable* due to differences such as length limit (100 words or 665 bytes) and evaluation metric (ROUGE F1 or recall). Details of each method and differences in evaluation can be found in App. A.

For extractive methods, we compare with SumBasic (Vanderwende et al., 2007), KL-Summ (Haghighi and Vanderwende, 2009), LexRank (Erkan and Radev, 2004), Centroid (Hong et al., 2014), ICSISumm (Gillick and Favre, 2009), rnn-ext + RL (Chen and Bansal, 2018), DPP (Kulesza and Taskar, 2012), and DPP-Caps-Comb (Cho et al., 2019). For abstractive methods, we compare with Opinosis (Ganesan et al., 2010), Extract+Rewrite (Song et al., 2018), PG (See et al., 2017), and PG-MMR (Lebanoff et al., 2018).

We use RL-MMR$_{\text{SOFT-ATTN}}$ as our default model unless otherwise mentioned. Implementation details can be found in App. A.4. We also report Oracle, an approximate upper bound that greedily extracts sentences to maximize ROUGE-1 F1 given the reference summaries (Nallapati et al., 2017).

4.2 Experimental Results

4.2.1 Comparison with the State-of-the-art

To answer Q1, we compare RL-MMR with state-of-the-art summarization methods and list the comparison results in Tables 1 and 2.

On DUC 2004, we observe that rnn-ext + RL, which we base our framework on, fails to achieve satisfactory performance even after fine-tuning. The large performance gains of RL-MMR over rnn-ext + RL demonstrates the benefits of guiding SDS models with MMR when adapting them to MDS. A similar conclusion is reached when comparing PG and PG-MMR. However, the hard cutoff

[3]Parameters of ROUGE: -2 4 -U -r 1000 -n 2 -l 100 -m.

Method	DUC 2004		
	R-1	R-2	R-SU4
Opinosis	27.07	5.03	8.63
Extract+Rewrite	28.90	5.33	8.76
SumBasic	29.48	4.25	8.64
KLSumm	31.04	6.03	10.23
LexRank	34.44	7.11	11.19
Centroid	35.49	7.80	12.02
ICSISumm	37.31	9.36	13.12
PG	31.43	6.03	10.01
PG-MMR	36.88	8.73	12.64
rnn-ext + RL (pre-train)	32.76	6.09	10.36
rnn-ext + RL (fine-tune)	35.93	8.60	12.53
DPP	38.10	9.14	13.40
DPP-Caps-Comb†	37.97	9.68	13.53
RL-MMR (ours)	**38.56**	**10.02**	**13.80**
Oracle	39.67	10.07	14.31

Table 1: **ROUGE F1 of compared methods on DUC 2004**. †We re-evaluate DPP-Caps-Comb (Cho et al., 2019) using author-released output as we found its results did not follow the 100-word length limit.

Method	TAC 2011		
	R-1	R-2	R-SU4
Opinosis	25.15	5.12	8.12
Extract+Rewrite	29.07	6.11	9.20
SumBasic	31.58	6.06	10.06
KLSumm	31.23	7.07	10.56
LexRank	33.10	7.50	11.13
PG	31.44	6.40	10.20
PG-MMR	37.17	10.92	14.04
rnn-ext + RL (pre-train)	33.45	7.37	11.28
rnn-ext + RL (fine-tune)	37.13	10.72	14.16
DPP	36.95	9.83	13.57
DPP-Caps-Comb†	37.51	11.04	14.16
RL-MMR (ours)	**39.65**	**11.44**	**15.02**
Oracle	42.44	13.85	16.90

Table 2: **Results of automatic evaluation (ROUGE F1) on TAC 2011**. †The output of DPP-Caps-Comb is again re-evaluated by limiting to 100 words.

in PG-MMR and the lack of in-domain fine-tuning lead to its inferior performance. DPP and DPP-Caps-Comb obtain decent performance but could not outperform RL-MMR due to the lack of end-to-end representation learning. Lastly, RL-MMR achieves new state-of-the-art results, approaching the performance of Oracle, which has access to the reference summaries, especially on ROUGE-2. We observe similar trends on TAC 2011 in which RL-MMR again achieves state-of-the-art performance. The improvement over compared methods is especially significant on ROUGE-1 and ROUGE-SU4.

4.2.2 Analysis of RL-MMR Combination

We answer Q2 by comparing the performance of various combination mechanisms for RL-MMR.

Performance Comparison. As shown in Table 3, RL-MMR$_{\text{HARD-COMB}}$ performs better than RL-MMR$_{\text{HARD-CUT}}$, showing the effectiveness of using MMR scores instead of degenerating them into binary values. We test RL-MMR$_{\text{SOFT-COMB}}$ with different β but it generally performs much worse than other variants, which implies that naively incorporating MMR into representation learning through weighted average may loosen the guidance of MMR, losing the benefits of both modules. Infusing MMR via soft attention of the action space performs the best, demonstrating the effectiveness of MMR guidance in RL-MMR$_{\text{SOFT-ATTN}}$ for sentence representation learning.

Combination	DUC 2004		
	R-1	R-2	R-SU4
HARD-CUT	38.19	9.26	13.43
HARD-COMB	38.45	9.35	13.64
SOFT-COMB	37.70	8.90	12.98
SOFT-ATTN	**38.56**	**10.02**	**13.80**

Table 3: **Comparison of RL-MMR variants** with different combination mechanisms.

Hard Cutoff vs. Soft Attention. We further compare the extracted summaries of MMR, RL-MMR$_{\text{HARD-CUT}}$, and RL-MMR$_{\text{SOFT-ATTN}}$ to verify the assumption that there are high-quality sentences not ranked highly by MMR and thus neglected by the hard cutoff. In our analysis, we find that when performing soft attention, 32% (12%) of extracted summaries contain low-ranked sentences that are not from $M^1_{1:K}$ when $K = 1$ ($K = 7$). We then evaluate those samples with low-ranked sentences extracted and conduct a pairwise comparison. On average, we observe a gain of 18.9% ROUGE-2 F1 of RL-MMR$_{\text{SOFT-ATTN}}$ over MMR, and 2.71% over RL-MMR$_{\text{HARD-CUT}}$, which demonstrates the benefits of soft attention.

Degree of RL-MMR Combination. To study the effect of RL-MMR combination in different degrees, we vary the cutoff K in RL-MMR$_{\text{HARD-CUT}}$ and analyze performance changes. As listed in Table 4, a small $K(=1)$ imposes tight constraints and practically degrades RL-MMR to vanilla MMR. A large $K(=50)$ might be too loose to limit the search space effectively, resulting in worse performance than a $K(=7, 10)$ within the proper range. When K is increased to 100, the impact of MMR further decreases but still positively influences model performance compared to the vanilla RL ($K = \infty$), especially on ROUGE-1.

K	DUC 2004			TAC 2011		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
1	37.91	8.83	13.10	38.54	10.83	14.43
7	38.19	**9.26**	13.43	**39.22**	**11.10**	**14.78**
10	**38.22**	9.24	**13.49**	39.13	11.07	14.63
50	38.12	9.23	13.42	38.60	11.05	14.55
100	36.92	8.98	12.98	37.94	10.92	14.20
∞	35.93	8.60	12.53	37.13	10.72	14.16

Table 4: **Performance changes of RL-MMR$_{\text{HARD-CUT}}$** when different cutoffs (K) are used.

4.2.3 Effectiveness of MMR Guidance

To answer Q3, we compare RL-MMR with vanilla RL without MMR guidance in terms of both training and test performance. We also inspect details such as runtime and quality of their extracted summaries (provided in App.B).

Training Performance. To examine whether MMR guidance helps with the learning efficiency of MDS, we plot the learning curves of vanilla RL and RL-MMR$_{\text{HARD-CUT}}$ in Fig. 2. RL-MMR receives a significantly better initial reward on the training set because MMR provides prior knowledge to extract high-quality sentences. In addition, RL-MMR has lower variance and achieves faster convergence than RL due to MMR guidance. Note that the final reward of vanilla RL on the plateau is higher than RL-MMR, which is somewhat expected since RL can achieve better fitting on the training set when it has less guidance (constraint).

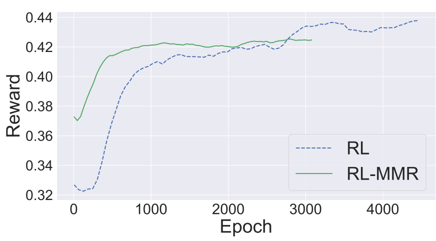

Figure 2: **The learning curves of RL and RL-MMR on DUC 2004**. RL training is more stable and converges faster when equipped with MMR.

Test Performance. We compare the test performance of vanilla RL and RL-MMR in Table 5. Despite the fact that vanilla RL obtains better training performance, its test performance is significantly worse than RL-MMR. Such a contradiction indicates that vanilla RL overfits on the training set and does not generalize well, again demonstrating the benefits of MMR guidance. We also find that, perhaps surprisingly, MMR outperforms vanilla RL even when RL is fine-tuned using in-domain training data. We thus believe that MMR and its methodology are underestimated by prior studies and should be explored further. Finally, RL-MMR achieves significantly better results than either RL or MMR alone, demonstrating the superiority of combining RL with MMR for MDS.

Method	DUC 2004			TAC 2011		
	R-1	R-2	R-SU4	R-1	R-2	R-SU4
RL (pre-train)	32.76	6.09	10.36	33.45	7.37	11.28
RL (fine-tune)	35.93	8.60	12.53	37.13	10.72	14.16
MMR	37.90	8.83	13.10	38.53	10.83	14.44
RL-MMR	**38.56**	**10.02**	**13.80**	**39.65**	**11.44**	**15.02**

Table 5: **Comparison of MMR, RL, and RL-MMR** further shows the effectiveness of MMR guidance.

4.2.4 Ablation of MMR

In this section, we conduct more ablation of MMR given its decent performance. We study the balance between salience and redundancy, and the performance of different similarity measures. Specifically, we use TF-IDF and BERT (Devlin et al., 2019) as the sentence (document) representation and measure cosine similarity in $\mathbf{S}(s_j, \mathcal{D})$ and $\mathbf{R}(s_j, e)$. We also explore whether a semantic textual similarity model, SNN (Latkowski, 2018), is more effective in measuring redundancy $\mathbf{R}(s_j, e)$ than TF-IDF. The TF-IDF features are estimated on the MDS datasets while the neural models are pre-trained on their corresponding tasks.

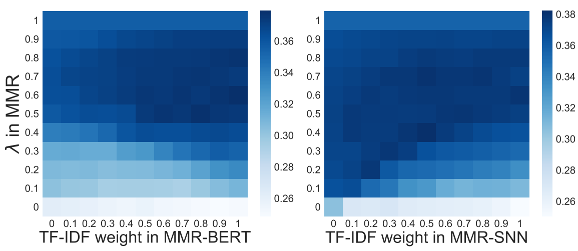

Figure 3: **Performance comparison under different salience-redundancy balances (y-axis), and various weighted combinations** of TF-IDF and BERT *(Left)* or SNN *(Right)* for similarity measure (x-axis). DUC 2004, ROUGE-1 F1, and $\mathbf{R}(s_j, e)$ are used for illustration. The results of other setups are similar.

Balance between Salience and Redundancy. By examining the y-axis in Fig. 3, we observe that con-

Reference Summary: PKK leader Ocalan was arrested on arrival at the Rome airport. He asked for asylum. Turkey pressured Italy to extradite Ocalan, whom they consider a terrorist. Kurds in Europe flocked to Rome to show their support. About 1,500 staged a hunger strike outside the hospital where he was held. Italy began a border crack-down to stop Kurds flocking to Rome. Greek media and officials oppose extradition. Romanian Kurds staged a 1-day business shutdown to protest his arrest. In a Turkish prison, an Italian prisoner was taken hostage. The Turkish president needed extra security for a trip to Austria. This is Italy's Prime Minister D'Alema's first foreign policy test.	**MMR**: 1. Turkey wants Italy to extradite the rebel, Abdullah Ocalan, leader of the Kurdistan Workers' Party, which is seeking Kurdish autonomy in southeastern Turkey. 2. Earlier Monday, while members of D'Alema's government met with Turkish officials who were in Rome for a European ministerial meeting, thousands of Kurds flooded into Rome to hold a demonstration and hunger strike in support of Ocalan. 3. The extra effort was prompted by the arrest last week in Rome of Abdullah Ocalan, the chief of the Turkish Workers Party PKK, Zehetmayr said. 4. If Italy sends the Kurd leader back to Turkey, he'll be tortured for certain, said Dino Frisullo, an Italian supporter among the singing, chanting Kurds outside the military hospital.
DPP-Caps-Comb: 1. Turkey has asked for his extradition and Ocalan has asked for political asylum. 2. Turkey stepped up the pressure on Italy for the extradition of captured Kurdish rebel leader Abdullah Ocalan, warning Sunday that granting him asylum would amount to opening doors to terrorism. 3. If Italy sends Ocalan back to Turkey, he'll be tortured for certain, said Dino Frisullo, an Italian supporter among the singing, chanting Kurds outside the military hospital. 4. Thousands of Kurds living in Romania closed down restaurants, shops and companies to protest the arrest of leader Abdullah Ocalan by Italian authorities, a newspaper reported Tuesday. 5. Turkey wants Italy to extradite the rebel, Abdullah Ocalan, leader of the Kurdistan Workers' Party, which is seeking Kurdish autonomy in southeastern Turkey.	**RL-MMR**: 1. Turkey wants Italy to extradite the rebel, Abdullah Ocalan, leader of the Kurdistan Workers' Party, which is seeking Kurdish autonomy in southeastern Turkey. 2. In Rome, 1,500 Kurds massed for a second day of demonstrations outside the military hospital where Ocalan is believed to be held. 3. Thousands of Kurds living in Romania closed down restaurants, shops and companies to protest the arrest of leader Abdullah Ocalan by Italian authorities, a newspaper reported Tuesday. 4. Greek media and officials leveled strong opposition Sunday to the possible extradition of Abdullah Ocalan, the arrested Kurdish guerrilla leader, to Greece's traditional rival Turkey.

Table 6: **System summaries of different methods**. Text spans matched (unmatched) with the reference summary are in blue (green). Redundant spans are in red. Spans of the reference covered by RL-MMR are also in blue.

sidering both salience and redundancy (best λ = 0.5 ~ 0.8) performs much better than only considering salience ($\lambda = 1$) regardless of the specific measures, further indicating the necessity of explicit redundancy avoidance in MDS.

Comparison of Similarity Measures. By varying the x values in Fig. 3, TF-IDF and neural estimations are combined using different weights. Although BERT and SNN (combined with TF-IDF) perform *slightly* better at times, they often require careful hyper-parameter tuning (both x and y). Hence, We use TF-IDF as the representation in MMR throughout our experiments.

4.2.5 Output Analysis

We analyze the outputs of the best-performing methods in Table 6. DPP-Caps-Comb still seems to struggle with redundancy as it extracts three sentences with similar semantics ("Turkey wants Italy to extradite Ocalan"). MMR and DPP-Caps-Comb both extract one sentence regarding a hypothesis that "Ocalan will be tortured", which is not found in the reference. RL-MMR has a more salient and non-redundant summary, as it is end-to-end trained with advances in SDS for sentence representation learning while maintaining the benefits of classical MDS approaches. In contrast, MMR alone only considers lexical similarity; The redundancy measure in DPP-Caps-Comb is pre-trained on one SDS dataset with weak supervision and fixed during the training of DPP.

5 Related Work

Multi-document Summarization. Classical MDS explore both extractive (Erkan and Radev, 2004; Haghighi and Vanderwende, 2009) and abstractive methods (Barzilay et al., 1999; Ganesan et al., 2010). Many neural MDS methods (Yasunaga et al., 2017; Zhang et al., 2018) are merely comparable or even worse than classical methods due to the challenges of large search space and limited training data. Unlike DPP-Caps-Comb (Cho et al., 2019) that incorporates neural measures into classical MDS as features, RL-MMR opts for the opposite by endowing SDS methods with the capability to conduct MDS, enabling the potential of further improvement with advances in SDS.

Bridging SDS and MDS. Initial trials adapting SDS models to MDS (Lebanoff et al., 2018; Zhang et al., 2018) directly reuse SDS models (See et al., 2017; Tan et al., 2017). To deal with the large search space, a sentence ranker is used in the adapted models for candidate pruning. Specifically, Lebanoff et al. (2018) leverages MMR (Carbonell and Goldstein, 1998) to rank sentences, allowing only the words in the top-ranked sentences to appear in the generated summary. Similarly, Zhang et al. (2018) uses topic-sensitive PageRank (Haveliwala, 2002) and computes attention only for the top-ranked sentences. Unlike RL-MMR, these adapted models use hard cutoff and (or) lack end-to-end training, failing to outperform state-of-the-art methods designed specifically for MDS (Gillick and Favre, 2009; Kulesza and Taskar, 2012).

6 Conclusion

We present a reinforcement learning framework for MDS that unifies neural SDS advances and Maximal Marginal Relevance (MMR) through end-to-end learning. The proposed framework leverages the benefits of both neural sequence learning and statistical measures, bridging the gap between SDS and MDS. We conduct extensive experiments on benchmark MDS datasets and demonstrate the superior performance of the proposed framework, especially in handling the large search space and high redundancy of MDS. In the future, we will investigate the feasibility of incorporating classical MDS guidance to abstractive models with large-scale pre-training (Gu et al., 2020) and more challenging settings where each document set may contain hundreds or even thousands of documents.

Acknowledgments

We thank Yu Zhang and Yu Meng for their help at the early stage of the work. Research was sponsored in part by US DARPA KAIROS Program No. FA8750-19-2-1004 and SocialSim Program No. W911NF-17-C-0099, National Science Foundation IIS-19-56151, IIS-17-41317, IIS 17-04532, and IIS 16-18481, and DTRA HDTRA11810026. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright annotation hereon.

References

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *27th Annual Meeting of the Association for Computational Linguistics, University of Maryland, College Park, Maryland, USA, 20-26 June 1999*, pages 550–557. ACL.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2017. Improving multi-document summarization via text classification. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3053–3059. AAAI Press.

Ziqiang Cao, Wenjie Li, Sujian Li, Furu Wei, and Yanran Li. 2016. Attsum: Joint learning of focusing and summarization with neural attention. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 547–556. ACL.

Jaime G Carbonell and Jade Goldstein. 1998. The use of mmr and diversity-based reranking for reodering documents and producing summaries.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 675–686. Association for Computational Linguistics.

Sangwoo Cho, Logan Lebanoff, Hassan Foroosh, and Fei Liu. 2019. Improving the similarity measure of determinantal point processes for extractive multi-document summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1027–1038. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.*, 22:457–479.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 340–348. Tsinghua University Press.

Yang Gao, Christian M. Meyer, Mohsen Mesgar, and Iryna Gurevych. 2019. Reward learning for efficient reinforcement learning in extractive document summarisation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 2350–2356. ijcai.org.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, pages 10–18, Boulder, Colorado. Association for Computational Linguistics.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Xiaotao Gu, Yuning Mao, Jiawei Han, Jialu Liu, You Wu, Cong Yu, Daniel Finnie, Hongkun Yu, Jiaqi Zhai, and Nicholas Zukoski. 2020. Generating representative headlines for news stories. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 1773–1784. ACM / IW3C2.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA*, pages 362–370. The Association for Computational Linguistics.

Taher H. Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, May 7-11, 2002, Honolulu, Hawaii, USA*, pages 517–526. ACM.

Geoffrey E. Hinton, Sara Sabour, and Nicholas Frosst. 2018. Matrix capsules with EM routing. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Kai Hong, John M. Conroy, Benoît Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 1608–1616. European Language Resources Association (ELRA).

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. 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 1746–1751. ACL.

Alex Kulesza and Ben Taskar. 2012. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2-3):123–286.

Tomasz Latkowski. 2018. Siamese deep neural networks for semantic similarity. `https://github.com/tlatkowski/multihead-siamese-nets`.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4131–4141. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Piji Li, Zihao Wang, Wai Lam, Zhaochun Ren, and Lidong Bing. 2017. Salience estimation via variational auto-encoders for multi-document summarization. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3497–3503. AAAI Press.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3728–3738. Association for Computational Linguistics.

Yuning Mao, Liyuan Liu, Qi Zhu, Xiang Ren, and Jiawei Han. 2020. Facet-aware evaluation for extractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4941–4957, Online. Association for Computational Linguistics.

Yuning Mao, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han. 2018. End-to-end reinforcement learning for automatic taxonomy induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2462–2472. Association for Computational Linguistics.

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019. Hierarchical text classification with reinforced label assignment. 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 2019, Hong Kong, China, November 3-7, 2019, pages 445–455. Association for Computational Linguistics.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3075–3081. AAAI Press.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1747–1759. Association for Computational Linguistics.

Mir Tafseer Nayeem, Tanvir Ahmed Fuad, and Yllias Chali. 2018. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1191–1204. Association for Computational Linguistics.

Karolina Owczarzak and Hoa Trang Dang. 2011. Overview of the tac 2011 summarization track: Guided task and aesop task. In *Proceedings of the Text Analysis Conference (TAC 2011), Gaithersburg, Maryland, USA, November*.

Over Paul and Yen James. 2004. An introduction to duc-2004. In *DUC*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Kaiqiang Song, Lin Zhao, and Fei Liu. 2018. Structure-infused copy mechanisms for abstractive summarization. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1717–1729. Association for Computational Linguistics.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1171–1181. Association for Computational Linguistics.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Inf. Process. Manag.*, 43(6):1606–1618.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order matters: Sequence to sequence for sets. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Kexiang Wang, Tianyu Liu, Zhifang Sui, and Baobao Chang. 2017. Affinity-preserving random walk for multi-document summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 210–220. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 985–992, Manchester, UK. Coling 2008 Organizing Committee.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir R. Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pages 452–462. Association for Computational Linguistics.

Jianmin Zhang, Jiwei Tan, and Xiaojun Wan. 2018. Adapting neural single-document summarization model for abstractive multi-document summarization: A pilot study. In *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, pages 381–390. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *ICML*, abs/1912.08777.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 563–578. Association for Computational Linguistics.

A Experimental Details

A.1 Dataset Statistics

We list in Table 7 the details of datasets used in our experiments. SDS methods usually take the first 256 or 512 words in the document as model input, which is infeasible for the input size of MDS (5,000 to 7,000 words on average).

Dataset	$\#\mathcal{D}$	$\sum\|\mathcal{D}\|$	$\overline{\sum\|D_i\|}$	$\min\sum\|D_i\|$	$\max\sum\|D_i\|$	$\overline{\sum\|s_j\|}$
DUC 2003	30	298	259.0	98	502	6830.5
DUC 2004	50	500	265.4	152	605	6987.1
TAC 2008-2010	138	1380	236.9	41	649	5978.4
TAC 2011	44	440	204.9	48	486	5146.0

Table 7: **Dataset statistics**. $\#\mathcal{D}$ and $\sum|\mathcal{D}|$ denote the number of document sets and the number of documents in total. $\overline{\sum|D_i|}$, $\min\sum|D_i|$, and $\max\sum|D_i|$ denote the average / min / max number of sentences in a document set. $\overline{\sum|s_j|}$ denotes the average number of words in a document set.

A.2 Remarks on Experimental Setup

We note that there are plenty of inconsistencies in the previous work on MDS and some results cannot be directly compared with ours. Specifically, there are three major differences that may lead to incomparable results as follows. First, while in the original DUC competitions an output length of 665 bytes is adopted, more recent studies mostly take a length limit of 100 words following Hong et al. (2014), and some do not have any length limit (usually resulting in higher numbers). Second, some papers report ROUGE recall (Yasunaga et al., 2017; Wang et al., 2017; Cao et al., 2017; Nayeem et al., 2018; Gao et al., 2019) while others (including ours) report ROUGE F1 following the trend on SDS (Lebanoff et al., 2018; Zhang et al., 2018; Cho et al., 2019). Third, while DUC 2004 and TAC 2011 are usually used as test sets, the training sets used in different studies often vary. We follow the same setup as the compared methods to ensure a fair comparison.

A.3 Description of Extractive Baselines

SumBasic (Vanderwende et al., 2007) is based on word frequency and hypothesizes that the words occurring frequently are likely to be included in the summary. KLSumm (Haghighi and Vanderwende, 2009) greedily extracts sentences as long as they can lead to a decrease in KL divergence. LexRank (Erkan and Radev, 2004) computes sentence salience based on eigenvector centrality in a graph-based representation. Centroid (Hong et al., 2014) measures sentence salience based on its cosine similarity with the document centroid, which is similar to the salience measure in MMR. ICSISumm (Gillick and Favre, 2009) uses integer linear programming (ILP) to extract a globally optimal set of sentences that can cover the most important concepts in the document set. DPP (Kulesza and Taskar, 2012) handles sentence salience and redundancy through the determinantal point processes, in which many handcrafted features such as sentence length, sentence position, and personal pronouns are used. DPP-Caps-Comb (Cho et al., 2019) improves upon DPP (Kulesza and Taskar, 2012) by replacing or combining the existing sentence salience and redundancy measures with capsule networks (Hinton et al., 2018). rnn-ext + RL (Chen and Bansal, 2018) is the SDS method that we base our work on. It is pre-trained on the CNN/Daily Mail SDS dataset (Nallapati et al., 2016), and we test its performance with or without fine-tuning on the MDS training set. The pre-trained abstractor in rnn-ext + RL is not used as we found it consistently leads to worse performance.

A.4 Description of Abstractive Baselines

Opinosis (Ganesan et al., 2010) generates summaries by finding salient paths on a word co-occurrence graph of the documents. Extract+Rewrite (Song et al., 2018) scores sentences by LexRank (Erkan and Radev, 2004) and employs an encoder-decoder model pre-trained on Gigaword (Graff et al., 2003) to generate a title-like summary for each sentence. PG (See et al., 2017) is one typical abstractive summarization method for SDS that conducts sequence-to-sequence learning with copy mechanism. PG-MMR (Lebanoff et al., 2018) adapts PG (See et al., 2017) to MDS by concatenating all of the documents in one document set and running pre-trained PG under the constraints of MMR on its vocabulary.

A.5 Implementation Details

Following common practice, we only consider extracting sentences with reasonable length (*i.e.*, 8 to 55 words) (Erkan and Radev, 2004; Yasunaga et al., 2017). We filter sentences that start with a quotation mark or do not end with a period (Wong et al., 2008; Lebanoff et al., 2018). For MMR, we set $\lambda = 0.6$ following Lebanoff et al. (2018). By default, we use TF-IDF features and cosine similarity for both sentence salience and redundancy measurement in MMR. We prefer such measurements

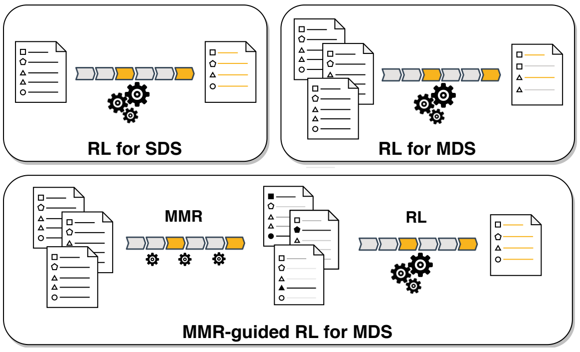

Figure 4: We use the same shape to denote semantically similar sentences. Directly applying RL to MDS encounters large search space and high redundancy, resulting in repeated summaries. MMR guides RL by attending to salient and non-redundant candidates.

instead of ROUGE-based measures (Lebanoff et al., 2018) and advanced neural-based measures (Cho et al., 2019; Devlin et al., 2019; Latkowski, 2018) as they are faster to compute and comparable in performance. We pre-train rnn-ext + RL (Chen and Bansal, 2018) on the CNN/Daily Mail SDS dataset (Nallapati et al., 2016) as in Lebanoff et al. (2018) but continue fine-tuning on the in-domain training set. We train RL-MMR using an Adam optimizer with learning rate 5e-4 for RL-MMR$_{\text{SOFT-ATTN}}$ and 1e-3 for the other variants without weight decay. we tested various reward functions, such as different ROUGE metrics, the MMR scores, and intrinsic measures based on sentence representation, and found them comparable or worse than the current one. One may also use other semantic metrics such as MoverScore (Zhao et al., 2019) and FAR (Mao et al., 2020).

B Detailed Analysis of RL-MMR

Additional Illustration. We provide an illustration in Fig. 4 to better elucidate the motivation of RL-MMR. Note that RL-MMR is mostly based on SDS architectures while achieving state-of-the-art performance on MDS, while existing combination approaches that achieve decent performance (*e.g.*, DPP-Caps) are based on MDS architectures.

Runtime and Memory Usage. RL-MMR is time and space efficient for two reasons. First, its hierarchical sentence encoding is much faster than a word-level sequence encoding mechanism while still capturing global context. Second, the guidance of MMR provides RL-MMR with a "warmup" effect, leading to faster convergence. In our experiments, one epoch of RL-MMR takes 0.87 to 0.91s on a GTX 1080 GPU with less than 1.2 GB memory usage. The number of epochs is set to 10,000 and we adopt early stopping – the training process terminates if RL-MMR cannot achieve better results on the validation set after 30 continuous evaluations. As a result, the runs often terminate before 5,000 epochs, and the overall training time ranges from 40 to 90 minutes.

Detailed Examples. In Table 8, we show the extracted summaries of vanilla RL and RL-MMR for the same document set. Without the guidance of MMR, the RL agent is much more likely to extract redundant sentences. In the first example, RL extracts two semantically equivalent sentences from two different documents. These two sentences would have similar sentence representation $\boldsymbol{h}_j^i$, and the latent state representation $\boldsymbol{g}_t$ itself might not be enough to avoid redundant extraction. In contrast, RL-MMR selects diverse sentences after extracting the same original sentence as RL thanks to the explicit redundancy measure in MMR. In the second example, the issue of redundancy in RL is even more severe – all four extracted sentences of RL are covering the same aspect of the news. RL-MMR again balances sentence salience and redundancy better than vanilla RL, favoring diverse sentences. Such results imply that pure neural representation is insufficient for redundancy avoidance in MDS and that classical approaches can serve as a complement.

RL: 1. President Clinton made an unusual, direct appeal to North Korea on Saturday to set aside any nuclear ambitions in favor of strengthening ties to South Korea and the United States.
2. SEOUL, South Korea (AP) U.S. President Bill Clinton won South Korea's support Saturday for confronting North Korea over a suspected nuclear site, and he warned the North's communist leaders not to squander **an historic chance to make a lasting peace** on the peninsula.
3. SEOUL, South Korea (AP) U.S. President Bill Clinton won South Korea's support Saturday for confronting North Korea over a suspected nuclear site, and he warned the North's communist leaders not to squander **a chance to achieve lasting peace** on the peninsula.

RL-MMR: 1. SEOUL, South Korea (AP) U.S. President Bill Clinton won South Korea's support ... an historic chance to make a lasting peace on the peninsula.
2. The North Koreans have denied that the complex, which is being built on a mountainside about 25 miles northeast of Yongbyon, the former North Korean nuclear research center, is intended to be used for a nuclear weapons program.
3. The United States and North Korea are set to resume talks Friday about inspections of an underground North Korean site suspected of being used to produce nuclear weapons.

RL: 1. Galina Starovoitova, 52, a leader of the liberal Russia's Democratic Choice party, **was shot dead** by unidentified assailants on the stairs of her apartment building in St. Petersburg on Friday night.
2. A liberal lawmaker who planned to run for president in Russia's next elections **was shot to death** Friday in St. Petersburg, **police said**.
3. A liberal lawmaker who planned to run for president in Russia's next elections **was shot to death** Friday in St. Petersburg, **police said**.
4. A liberal lawmaker who planned to run for president in Russia's next elections **was killed** Friday in St. Petersburg, **a news report said**.

RL-MMR: 1. Galina Starovoitova, 52, a leader of the liberal Russia's Democratic Choice party, was shot dead by unidentified assailants on the stairs of her apartment...
2. Starovoitova tried to run for president in the 1996 elections but her registration was turned down for technical reasons.
3. Like that fictional crime, which shone a light on social ferment in the St. Petersburg of its day, the death of Starovoitova was immediately seized upon as a seminal event in the Russia of the late 1990s.
4. She was a member of the Russian parliament and a recently declared candidate for governor of the region around St. Petersburg.

Table 8: **Case studies reveal the insufficient redundancy measure in vanilla RL**. Note that the 2nd and 3rd extracted sentences of RL in the second example are the same but from different documents, which is quite typical in news reports.