Adapting the Neural Encoder-Decoder Framework from Single to Multi-Document Summarization

Anonymous EMNLP submission

Abstract

Generating an abstract from a set of relevant documents remains challenging. Despite the development of the neural encoder-decoder framework, prior studies focus primarily on single-document summarization, possibly because labelled training data can be automatically harvested from the Web. Nevertheless, labelled data for multi-document summarization are scarce. There is thus an increasing need to adapt the encoder-decoder framework from single- to multiple-document summarization in an unsupervised fashion. In this paper we present an initial investigation into a novel adaptation method. It exploits the maximal marginal relevance method to select representative sentences from multi-document input, and an abstractive encoder-decoder model to fuse disparate sentences to an abstractive summary. The adaptation method is robust and itself requires no training data. Our system compares favorably to state-of-the-art extractive and abstractive approaches judged by both automatic metrics and human assessors.

1 Introduction

Neural abstractive summarization has primarily focused on summarizing short texts written by single authors. For example, *sentence summarization* seeks to reduce the first sentence of a news article to a title-like summary (Rush et al., 2015; Nallapati et al., 2016; Takase et al., 2016); *single-document summarization* (**SDS**) focuses on condensing a full news article to a handful of bullet points (Paulus et al., 2017; See et al., 2017). These summarization studies are empowered by the large parallel datasets automatically harvested from major news outlets, including the Gigaword (Rush et al., 2015), CNN/Daily Mail (Hermann et al., 2015), and NYT (Sandhaus, 2008) datasets.

To date, *multi-document summarization* (**MDS**) has not yet fully benefited from the development of the neural encoder-decoder models. MDS seeks to condense a set of relevant documents written by multiple authors to a short and concise summary. It has practical applications, such as summarizing product reviews (Gerani et al., 2014), student responses to post-class questionnaires (Luo and Litman, 2015), and sets of news articles discussing particular topics (Hong et al., 2014). State-of-the-art MDS systems are largely extractive (Nenkova and McKeown, 2011). Despite their promising results, such systems cannot perform text abstraction, e.g., paraphrasing, generalization, and sentence fusion (Jing and McKeown, 1999). Further, annotated MDS datasets are often scarce, containing only hundreds of training pairs (see Table 1). The cost to create ground-truth summaries from multiple-document inputs can be prohibitive. The MDS datasets are thus too small to be used to train neural encoder-decoder models with millions of parameters without overfitting.

DATASET	SOURCE	SUMMARY	#PAIRS
Gigaword (Rush et al., 2015)	**the first sentence** of a news article	**8.3 words** title-like	**4 Million**
CNN/Daily Mail (Hermann et al., 2015)	**a full article**	**56 words** multi-sent	**312 K**
TAC (08-11) (Dang et al., 2008)	**10 news articles** related to a topic	**100 words** multi-sent	**728**
DUC (03-04) (Over and Yen, 2004)	**10 news articles** related to a topic	**100 words** multi-sent	**320**

Table 1: A comparison of datasets for sentence summarization (Gigaword), single-doc (CNN/DM) and multi-doc summarization (DUC/TAC). The labelled data for multi-doc summarization are much less.

A promising route to generating an abstractive summary from the multi-document input is to apply a neural encoder-decoder model trained for single-document summarization to a "mega-document"—created by concatenating all input documents—at test time. However, the model may not scale well for two reasons. First, identifying important text pieces from the mega-document can be challenging. The encoder-decoder model is trained on single documents where summary content often appears in the first few sentences of a

document. However, this is not the case for the mega-document. Second, redundant text pieces in the mega-document can be repeatedly used for summary generation under the current framework. The attention mechanism of the encoder-decoder model (Bahdanau et al., 2014) is position-based and lacks an awareness of semantics. If a text piece has been attended to during summary generation, it is unlikely to be used again. However, the attention weight of a similar text piece in a different position is not affected. The same content can thus be repeatedly used for summary generation. These issues may be alleviated by improving the encoder-decoder architecture and its attention mechanism (Cheng and Lapata, 2016; Tan et al., 2017). However, in these cases the model has to be re-trained on large-scale MDS datasets that are not available at the current stage. There is thus an increasing need for a lightweight adaptation of an encoder-decoder model trained on SDS datasets to process multi-document inputs at test time.

In this paper we present a novel adaptation method, named PG-MMR, to generate abstracts from multi-document inputs. The method is robust and itself does not require training data. It combines a recent neural encoder-decoder model (PG for Pointer-Generator networks; See et al., 2017) that generates abstractive summaries from single-document inputs with a strong extractive summarization algorithm (MMR for Maximal Marginal Relevance; Carbonell and Goldstein, 1998) that identifies important source sentences from multi-document inputs. The PG-MMR system iteratively performs the following. It identifies a handful of the most important sentences from the mega-document. The word attention values of the PG model are directly modified to focus on the corresponding important sentences. Next, the system re-identifies a number of important sentences, but the likelihood of choosing certain sentences is reduced based on their similarity to the partially-generated summary, thereby reducing redundancy. Our research contributions include the following:

- we present an investigation into a novel adaptation method of the encoder-decoder framework from single to multi-document summarization. To the best of our knowledge, this is the first attempt at coupling the maximal marginal relevance algorithm with pointer-generator networks for multi-document summarization;
- we demonstrate the effectiveness of the proposed method through extensive experiments on standard MDS datasets. Our system compares favorably to state-of-the-art extractive and abstractive summarization systems measured by both automatic metrics and human judgments.

2 Related Work

Popular methods for multi-document summarization have been extractive. Important sentences are extracted from a set of relevant documents and optionally compressed to form a summary (Daume III and Marcu, 2002; Zajic et al., 2007; Gillick and Favre, 2009; Galanis and Androutsopoulos, 2010; Berg-Kirkpatrick et al., 2011; Thadani and McKeown, 2013; Wang et al., 2013; Hong et al., 2014; Filippova et al., 2015; Durrett et al., 2016). In recent years neural networks have been exploited to learn word/sentence representations for single and multi-document summarization (Cheng and Lapata, 2016; Cao et al., 2017; Isonuma et al., 2017; Yasunaga et al., 2017; Narayan et al., 2018). However, these approaches remain extractive. Despite the promising results, summarizing a large quantity of texts still requires sophisticated abstraction capabilities such as generalization, paraphrasing and sentence fusion.

Prior to the deep learning era, abstractive summarization has been studied (Barzilay et al., 1999; Carenini and Cheung, 2008; Ganesan et al., 2010; Gerani et al., 2014; Fabbrizio et al., 2014; Pighin et al., 2014; Bing et al., 2015). These approaches construct domain templates using a text planner or an open-IE system and employ a natural language generator for surface realization. Limited by the availability of labelled data, experiments are often performed on small domain-specific datasets.

Neural abstractive summarization utilizing the encoder-decoder architecture has shown promising results but studies focus primarily on sentence summarization and single-document summarization (Nallapati et al., 2016; Kikuchi et al., 2016; Chen et al., 2016; Miao and Blunsom, 2016; Tan et al., 2017; Zeng et al., 2017; Zhou et al., 2017; Paulus et al., 2017; See et al., 2017). In particular, the pointing mechanism (Gulcehre et al., 2016; Gu et al., 2016) enables the system to both copy words from the source text and generate new words from the vocabulary. Reinforcement learning is also exploited to directly optimize the evaluation metrics (Paulus et al., 2017; Chen and Bansal, 2018). These studies focus on summarizing sentences and single documents because the training data are abundant. To date, little research has been dedicated to investigate the feasibility of extending the encoder-decoder model to generate abstractive

summaries for multi-document inputs, where the available training data are scarce.

This paper presents some first steps towards this goal. We introduce a fully unsupervised adaptation method combining the pointer-generator (PG) networks (See et al., 2017) and the maximal marginal relevance (MMR) algorithm (Carbonell and Goldstein, 1998). The PG model, trained on SDS data and detailed in Section §3, is capable of generating document abstracts by performing text abstraction and sentence fusion. However, if the model is applied at test time to summarize multi-document inputs, there will be limitations. Our PG-MMR algorithm, presented in Section §4, teaches the PG model to effectively recognize important content from the input documents, hence improving the quality of abstractive summaries.

3 Limits of the Encoder-Decoder Model

The encoder-decoder architecture has become the *de facto* standard for neural abstractive summarization (Rush et al., 2015). The encoder is often a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) converting the input text to a set of hidden states $\{\mathbf{h}_i^e\}$, one for each input word, indexed by i. The decoder is a unidirectional LSTM that generates a summary by predicting one word at a time. The decoder hidden states are represented by $\{\mathbf{h}_t^d\}$, indexed by t. For sentence and single-document summarization (Nallapati et al., 2016; Paulus et al., 2017; See et al., 2017), the input text is treated as a sequence of words, and the model is expected to capture the source syntax inherently.

$$e_{t,i} = \mathbf{v}^\top \tanh(\mathbf{W}^e[\mathbf{h}_t^d||\mathbf{h}_i^e||\widetilde{\alpha}_{t,i}] + \mathbf{b}^e) \quad (1)$$

$$\alpha_{t,i} = \text{softmax}(e_{t,i}) \quad (2)$$

$$\widetilde{\alpha}_{t,i} = \textstyle\sum_{t'=0}^{t-1} \alpha_{t',i} \quad (3)$$

The attention weight $\alpha_{t,i}$ measures how important the i-th input word is to generating the t-th output word (Eq. (1-2)). Following (See et al., 2017), $\alpha_{t,i}$ is calculated by measuring the strength of interaction between the decoder hidden state $\mathbf{h}_t^d$, the encoder hidden state $\mathbf{h}_i^e$, and the *cumulative* attention $\widetilde{\alpha}_{t,i}$. $\widetilde{\alpha}_{t,i}$ denotes the cumulative attention that the i-th input word receives up to time step t-1 (Eq. (3)). A large value of $\widetilde{\alpha}_{t,i}$ indicates the i-th input word has been used prior to time t and it is unlikely to be used again for generating the t-th output word.

A context vector ($\mathbf{c}_t$) is constructed (Eq. (4)) to summarize the semantic meaning of the input; it is a weighted sum of the encoder hidden states. The context vector and the decoder hidden state ($[\mathbf{h}_t^d||\mathbf{c}_t]$) are then used to compute the vocabulary probability $P_{vcb}(w)$ measuring the likelihood of a vocabulary word w being selected as the t-th output word (Eq. (5)).[1]

$$\mathbf{c}_t = \textstyle\sum_i \alpha_{t,i}\mathbf{h}_i^e \quad (4)$$

$$P_{vcb}(w) = \text{softmax}(\mathbf{W}^y[\mathbf{h}_t^d||\mathbf{c}_t] + \mathbf{b}^y) \quad (5)$$

In many encoder-decoder models, a "switch" is estimated ($p_{gen} \in [0,1]$) to indicate whether the system has chosen to select a word from the vocabulary or to copy a word from the input text (Eq. (6)). The switch is computed using a feedforward layer with σ activation over $[\mathbf{h}_t^d||\mathbf{c}_t||\mathbf{y}_{t-1}]$, where $\mathbf{y}_{t-1}$ is the embedding of the output word at time t-1. The attention weights ($\alpha_{t,i}$) are used to compute the copy probability (Eq. (7)). If a word w appears once or more in the input text, its copy probability ($\sum_{i:w_i=w} \alpha_{t,i}$) is the sum of the attention weights over all its occurrences. The final probability $P(w)$ is a weighted combination of the vocabulary probability and the copy probability. A cross-entropy loss function can be used to train the model end-to-end.

$$p_{gen} = \sigma(\mathbf{w}^z[\mathbf{h}_t^d||\mathbf{c}_t||\mathbf{y}_{t-1}]) + b^z) \quad (6)$$

$$P(w) = p_{gen}P_{vcb}(w) + (1 - p_{gen}) \sum_{i:w_i=w} \alpha_{t,i} \quad (7)$$

To thoroughly understand the aforementioned encoder-decoder model, we divide its model parameters into four groups. They include

- parameters of the encoder and the decoder;
- $\{\mathbf{w}^z, b^z\}$ for calculating the "switch" (Eq. (6));
- $\{\mathbf{W}^y, \mathbf{b}^y\}$ for calculating $P_{vcb}(w)$ (Eq. (5));
- $\{\mathbf{v}, \mathbf{W}^e, \mathbf{b}^e\}$ for attention weights (Eq. (1)).

By training the encoder-decoder model on single-document summarization (SDS) data containing a large collection of news articles paired with summaries (Hermann et al., 2015), these model parameters can be effectively learned.

However, at test time, we wish for the model to generate abstractive summaries from *multi-document inputs*. This brings up two issues. First, the parameters are ineffective at identifying salient content from multi-document inputs. Humans are very good at identifying representative sentences from a set of documents and fusing them into an

[1] Here $[\cdot||\cdot]$ represents the concatenation of two vectors. The pointer-generator networks (See et al., 2017) use two linear layers to produce the vocabulary distribution $P_{vcb}(w)$. We use $\mathbf{W}^y$ and $\mathbf{b}^y$ to denote parameters of both layers.

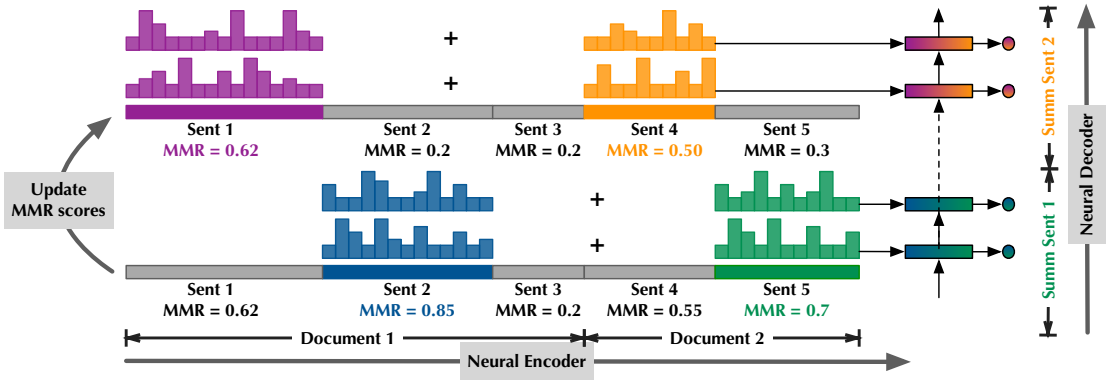

Figure 1: System framework. The PG-MMR system uses K highest-scored source sentences (in this case, K=2) to guide the PG model to generate a summary sentence. All other source sentences are "muted" in this process. Best viewed in color.

abstract. However, this capability is not supported by the encoder-decoder model. Second, the attention mechanism is based on input word positions but not their semantics. It can lead to redundant content in the multi-document input being repeatedly used for summary generation. We conjecture that both aspects can be addressed by introducing an "external" model that selects representative sentences from multi-document inputs and dynamically adjusts the sentence importance to reduce summary redundancy. This external model is integrated with the encoder-decoder model to generate abstractive summaries using selected representative sentences. In the following section we present our fully unsupervised adaptation method for multi-document summarization.

4 Our Method

Maximal marginal relevance. Our adaptation method combines the maximal marginal relevance algorithm (MMR; Carbonell and Goldstein, 1998) and the pointer-generator networks (PG; See et al., 2017) to generate abstractive summaries from sets of multiple documents. MMR is one of the most successful extractive approaches and, despite its straightforwardness, performs on-par with state-of-the-art systems (Luo and Litman, 2015; Yogatama et al., 2015). At each iteration, MMR selects one sentence from the document (D) and includes it in the summary (S) until a length threshold is reached. The selected sentence (s_i) is the most important one among the remaining sentences and it has the least content overlap with the current summary. In the equation below, $\text{Sim}_1(s_i, D)$ measures the similarity of the sentence s_i to the document. It serves as a proxy of sentence importance, since important sentences usually show similarity to the centroid of the document. $\max_{s_j \in S} \text{Sim}_2(s_i, s_j)$ measures the maximum similarity of the sentence s_i to each of the summary sentences, acting as a proxy of redundancy. λ is a balancing factor.

$$\underset{s_i \in D \setminus S}{\text{argmax}} \Big[\underbrace{\lambda \text{Sim}_1(s_i, D)}_{\text{importance}} - \underbrace{(1-\lambda) \max_{s_j \in S} \text{Sim}_2(s_i, s_j)}_{\text{redundancy}} \Big]$$

Our PG-MMR describes an iterative framework for summarizing a multi-document input to a summary consisting of multiple sentences. At each iteration, PG-MMR follows the MMR principle to select the K highest-scored source sentences; they serve as the basis for PG to generate a summary sentence. After that, the scores of all source sentences are updated based on their importance and redundancy. Sentences that are highly similar to the partial summary receive lower scores. Selecting K sentences via the MMR algorithm helps the PG system to effectively identify salient source content that has not been included in the summary.

Muting. To allow the PG system to effectively utilize the K source sentences without retraining the neural model, we dynamically adjust the PG attention weights ($\alpha_{t,i}$) at test time. Let S_k represent a selected sentence. The attention weights of the words belonging to $\{\text{S}_k\}_{k=1}^{K}$ are calculated as before (Eq. (2)). However, words in other sentences are forced to receive zero attention weights ($\alpha_{t,i}$=0), and all $\alpha_{t,i}$ are renormalized (Eq. (8)).

$$\alpha_{t,i}^{\text{new}} = \begin{cases} \alpha_{t,i} & i \in \{\text{S}_k\}_{k=1}^{K} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

It means that the remaining sentences are "muted" in this process. In this variant, the sentence importance does not affect the original attention weights, other than muting.

In an alternative setting, the sentence salience is multiplied with the word salience and renormalized (Eq. (9)). PG uses the reweighted alpha values to predict the next summary word.

$$\alpha_{t,i}^{\text{new}} = \begin{cases} \alpha_{t,i}\text{MMR}(\text{S}_k) & i \in \{\text{S}_k\}_{k=1}^{K} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Sentence Importance. To estimate sentence importance $\text{Sim}_1(s_i, D)$, we introduce a supervised regression model in this work. Importantly, the model is trained on single-document summarization datasets where training data are abundant. At test time, the model can be applied to identify important sentences from multi-document input. Our model determines sentence importance based on four indicators, inspired by how humans identify important sentences from a document set. They include (a) sentence length, (b) its absolute and relative position in the document, (c) sentence quality, and (d) how close the sentence is to the main topic of the document set. These features are considered to be important indicators in previous extractive summarization framework (Galanis and Androutsopoulos, 2010; Hong et al., 2014).

Regarding the sentence quality (c), we leverage the PG model to build the sentence representation. We use the bidirectional LSTM encoder to encode any source sentence to a vector representation. $[\overrightarrow{\mathbf{h}_D^e}||\overleftarrow{\mathbf{h}_1^e}]$ is the concatenation of the last hidden states of the forward and backward passes. A document vector is the average of all sentence vectors. We use the document vector and the cosine similarity between the document and sentence vectors as indicator (d). A support vector regression model is trained on (sentence, score) pairs where the training data are obtained from the CNN/Daily Mail dataset. The target importance score is the ROUGE-L recall of the sentence compared to the ground-truth summary. Our model leverages neural representations of sentences and documents, they are data-driven and not specifically designed for a particular domain.

Sentence Redundancy. To calculate the redundancy of the sentence ($\max_{s_j \in S} \text{Sim}_2(s_i, s_j)$), we compute the ROUGE-L precision, which measures the longest common subsequence between a source sentence and the partial summary (consisting of all sentences generated thus far by the PG model), divided by the length of the source sentence. A source sentence yielding a high ROUGE-L precision is deemed to have significant content overlap with the partial summary. It will receive a low MMR score and hence is less likely to serve as basis for generating future summary sentences.

Algorithm 1 The PG-MMR algorithm for summarizing multi-document inputs.

Input: SDS data; MDS source sentences $\{\text{S}_i\}$.

Train the PG model on SDS data.
▶ $\mathcal{I}(\text{S}_i)$ and $\mathcal{R}(\text{S}_i)$ are the importance and redundancy scores of the source sentence S_i.
$\text{MMR}(\text{S}_i) \leftarrow \lambda\mathcal{I}(\text{S}_i)$ for all source sentences.
Summary $\leftarrow \{\}$
$t \leftarrow$ index of summary words
while $t < L_{\text{max}}$ **do**
  Find $\{\text{S}_k\}_{k=1}^{K}$ with highest MMR scores.
  Compute $\alpha_{t,i}^{\text{new}}$ based on $\{\text{S}_k\}_{k=1}^{K}$ (Eq. (8))
  Run PG decoder for one step to get $\{w_t\}$.
  Summary $\leftarrow$ Summary + $\{w_t\}$
  if w_t is the period symbol **then**
    $\mathcal{R}(\text{S}_i) \leftarrow \text{Sim}(\text{S}_i, \text{Summary}), \forall i$
    $\text{MMR}(\text{S}_i) \leftarrow \lambda\mathcal{I}(\text{S}_i) - (1-\lambda)\mathcal{R}(\text{S}_i), \forall i$
  end if
end while

Alg. 1 provides an overview the PG-MMR algorithm and Fig. 1 is a graphical illustration. The MMR scores of source sentences are updated after each summary sentence is generated by the PG model. Next, a different set of highest-scored sentences are used to guide the PG model to generate the next summary sentence. "Muting" the remaining source sentences is important because it helps the PG model to focus its attention on the most significant source content. The code for our model will be shared publicly to further MDS research.

5 Experimental Setup

Datasets. We investigate the effectiveness of the PG-MMR method by testing it on standard multi-document summarization datasets (Over and Yen, 2004; Dang and Owczarzak, 2008). These include DUC-03, DUC-04, TAC-08, TAC-10, and TAC-11, containing 30/50/48/46/44 topics respectively. The summarization system is tasked with generating a concise, fluent summary of 100 words or less from a set of 10 documents discussing a topic. All documents in a set are chronologically ordered and concatenated to form a mega-document serving as input to the PG-MMR system. Sentences that start with a quotation mark or do not end with a period are excluded (Wong et al., 2008). Each system summary is compared against 4 human abstracts created by NIST assessors. Following convention, we report results on DUC-04 and TAC-11

datasets, which are standard test sets; DUC-03 and TAC-08/10 are used as a validation set for hyper-parameter tuning.

The PG model is trained for single-document summarization using the CNN/Daily Mail (Hermann et al., 2015) dataset, containing single news articles paired with summaries (human-written article highlights). The training set contains 287,226 articles. An article contains 781 tokens on average; and a summary contains 56 tokens (3.75 sentences). During training we use the hyperparameters provided by See et al. (2017). At test time, the maximum/minimum decoding steps are set to 120/100 words respectively, corresponding to the max/min lengths of the PG-MMR summaries. Because the focus of this work is on multi-document summarization (MDS), we do not report results for the CNN/Daily Mail dataset.

Baselines. We compare PG-MMR against a broad spectrum of baselines, including state-of-the-art extractive ('*ext*-') and abstractive ('*abs*-') systems. They are described below.[2]

- *ext*-**SumBasic** (Vanderwende et al., 2007) is an extractive approach assuming words occurring frequently in a document set are more likely to be included in the summary;
- *ext*-**KL-Sum** (Haghighi and Vanderwende, 2009) greedily adds source sentences to the summary if it leads to a decrease in KL divergence;
- *ext*-**LexRank** (Erkan and Radev, 2004) uses a graph-based approach to compute sentence importance based on eigenvector centrality in a graph representation;
- *ext*-**Centroid** (Hong et al., 2014) computes the importance of each source sentence based on its cosine similarity with the document centroid;
- *ext*-**ICSISumm** (Gillick et al., 2009) leverages the ILP framework to identify a globally-optimal set of sentences covering the most important concepts in the document set;
- *ext*-**DPP** (Taskar, 2012) selects an optimal set of sentences per the determinantal point processes that balance the coverage of important information and the sentence diversity;
- *abs*-**Opinosis** (Ganesan et al., 2010) generates abstractive summaries by searching for salient paths on a word co-occurrence graph created from source documents;
- *abs*-**Extract+Rewrite** (Anonymized) is a recent approach that scores sentences using LexRank and generates a title-like summary for each sentence using an encoder-decoder model trained on Gigaword (Rush et al., 2015).
- *abs*-**PG-Original** (See et al., 2017) introduces an encoder-decoder model that encourages the system to copy words from the source text via pointing, while retaining the ability to produce novel words through the generator.

[2]We are grateful to Hong et al. (2014) for providing us the summaries generated by Centroid, ICSISumm, DPP systems. These are only available for the DUC-04 dataset.

System	DUC-04 R-1	R-2	R-SU4
SumBasic (Vanderwende et al., 2007)	29.48	4.25	8.64
KLSumm (Haghighi et al., 2009)	31.04	6.03	10.23
LexRank (Erkan and Radev, 2004)	34.44	7.11	11.19
Centroid (Hong et al., 2014)	35.49	7.80	12.02
ICSISumm (Gillick and Favre, 2009)	37.31	9.36	13.12
DPP (Taskar, 2012)	38.78	9.47	13.36
Extract+Rewrite (Anonymized)	28.90	5.33	8.76
Opinosis (Ganesan et al., 2010)	27.07	5.03	8.63
PG-Original (See et al., 2017)	31.43	6.03	10.01
PG-MMR w/ SummRec	34.57	7.46	11.36
PG-MMR w/ SentAttn	36.52	8.52	12.57
PG-MMR w/ Cosine (*default*)	36.88	**8.73**	12.64
PG-MMR w/ BestSummRec	36.42	**9.36**	13.23

Table 2: ROUGE results on the DUC-04 dataset.

System	TAC-11 R-1	R-2	R-SU4
SumBasic (Vanderwende et al., 2007)	31.58	6.06	10.06
KLSumm (Haghighi et al., 2009)	31.23	7.07	10.56
LexRank (Erkan and Radev, 2004)	33.10	7.50	11.13
Extract+Rewrite (Anonymized)	29.07	6.11	9.20
Opinosis (Ganesan et al., 2010)	25.15	5.12	8.12
PG-Original (See et al., 2017)	31.44	6.40	10.20
PG-MMR w/ SummRec	35.06	8.72	12.39
PG-MMR w/ SentAttn	37.01	10.43	13.85
PG-MMR w/ Cosine (*default*)	37.17	**10.92**	14.04
PG-MMR w/ BestSummRec	40.44	**14.93**	17.61

Table 3: ROUGE results on the TAC-11 dataset.

6 Results

Having described the experimental setup, we next compare the PG-MMR method against the baselines on standard MDS datasets, evaluated by both automatic metrics and human assessors.

ROUGE (Lin, 2004). This automatic metric measures the overlap of unigrams (R-1), bigrams (R-2) and skip bigrams with a maximum distance of 4 words (R-SU4) between the system summary and a set of reference summaries. ROUGE scores of various systems are presented in Table 2 and 3 respectively for the DUC-04 and TAC-11 datasets.

We explore variants of the PG-MMR method. They differ in how the importances of source sentences are estimated and how the sentence importance affects word attention weights. "***w/ Cosine***" computes the sentence importance as the cosine similarity score between the sentence and document vectors, both represented as sparse TF-IDF vectors under the vector space model. "***w/ SummRec***" estimates the sentence importance as the predicted R-L recall score between the sentence and the summary. A support vector regression model is trained on sentences from the CNN/Daily Mail datasets (≈33K) and applied to DUC/TAC sentences at test time (see §4). "***w/ BestSumm-***

System	1-grams	2-grams	3-grams	Sent
Extr+Rewrite	89.37	54.34	25.10	6.65
PG-Original	99.64	96.28	88.83	47.67
PG-MMR	99.74	97.64	91.57	59.13
Human Abst.	84.32	45.22	18.70	0.23

Table 4: Percentages of summary n-grams (or the entire sentences) appear in the multi-document input. (TAC-11)

Rec" obtains the best estimate of sentence importance by calculating the R-L recall score between the sentence and reference summaries. It serves as an upper bound for the performance of "w/ SummRec." For all variants, the sentence importance scores are normalized to the range of [0,1]. "***w/ SentAttn***" adjusts the attention weights using Eq. (9), so that words in important sentences are more likely to be used to generate the summary. The weights are otherwise computed using Eq. (8).

As seen in Table 2 and 3, our PG-MMR method surpasses all unsupervised extractive baselines, including SumBasic, KLSumm, and LexRank. On the DUC-04 dataset, ICSISumm and DPP show good performance, but these systems are trained directly on MDS datasets, which are not utilized by the PG-MMR method. PG-MMR exhibits superior performance compared to existing abstractive systems. It outperforms Opinosis and PG-Original by a large margin in terms of R-2 F-scores (5.03/6.03/**8.73** for DUC-04 and 5.12/6.40/**10.92** for TAC-11). In particular, ***PG-Original*** is the original pointer-generator networks with multi-document inputs at test time. Compared to it, PG-MMR is more effective at identifying summary-worthy content from the input. "w/ Cosine" is used as the default PG-MMR and it shows better results than "w/ SummRec." It suggests that the sentence and document representations obtained from the encoder-decoder model (trained on CNN/DM) are suboptimal, possibly due to a vocabulary mismatch, where certain words in the DUC/TAC datasets do not appear in CNN/DM and their embeddings are thus not learned during training. Finally, we observe that "w/ BestSummRec" yields the highest performance on both datasets. This finding suggests that there is a great potential for improvements of the PG-MMR method as its "extractive" and "abstractive" components can be separately optimized.[3]

Location of summary content. We are interested in understanding why PG-MMR outperforms PG-Original at identifying summary content from the multi-document input. We ask the question: where, in the source documents, does each system tend to look when generating their summaries? Our findings indicate that PG-Original gravitates towards early source sentences, while PG-MMR searches beyond the first few sentences.

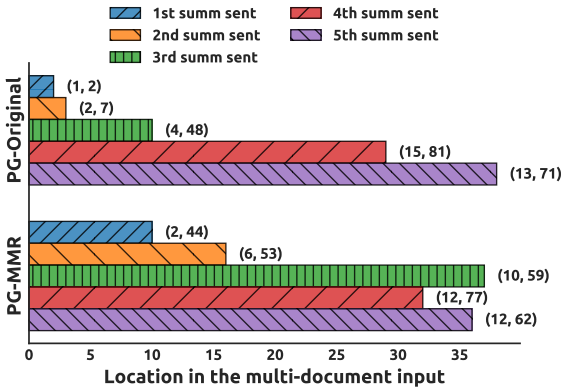

Figure 2: The median location of summary n-grams in the multi-document input (and the lower/higher quartiles). The n-grams come from the 1st/2nd/3rd/4th/5th summary sentence and the location is the source sentence index. (TAC-11)

In Figure 2 we show the median location of the first occurrences of summary n-grams, where the n-grams can come from the 1st to 5th summary sentence. For PG-Original summaries, n-grams of the 1st summary sentence frequently come from the 1st and 2nd source sentences, corresponding to the lower/higher quartiles of source sentence indices. Similarly, n-grams of the 2nd summary sentence come from the 2nd to 7th source sentences. For PG-MMR summaries, the patterns are different. The n-grams of the 1st and 2nd summary sentences come from source sentences of the range (2, 44) and (6, 53), respectively. Our findings suggest that PG-Original tends to treat the input as a single-document and identifies summary-worthy content from the beginning of the input, whereas PG-MMR can successfuly search a broader range of the input for summary content. This capability is crucial for multi-document input where important content can come from any article in the set.

Degree of extractiveness. Table 4 shows the percentages of summary n-grams (or entire sentences) appearing in the multi-document input. PG-Original and PG-MMR summaries both show a high degree of extractiveness, and similar findings have been revealed by See et al. (2017). Because PG-MMR relies on a handful of representative source sentences and mutes the rest, it appears to be marginally more extractive than PG-Original. Both systems encourage generating summary sentences by stitching together source sentences, as about 52% and 41% of the sum-

[3]The hyperparameters for all PG-MMR variants are K=7 and λ=0.6; except for "w/ BestSummRec" where K=2.

System	Linguistic Quality			Rankings (%)			
	Fluency	Inform.	NonRed.	1st	2nd	3rd	4th
Extract+Rewrite	2.03	2.19	1.88	5.6	11.6	11.6	**71.2**
LexRank	**3.29**	3.36	3.30	30.0	28.8	32.0	9.2
PG-Original	3.20	3.30	3.19	29.6	26.8	**32.8**	10.8
PG-MMR	3.24	**3.52**	**3.42**	**34.8**	**32.8**	23.6	8.8

Table 5: Linguistic quality and rankings of system summaries. (DUC-04)

Human Abstract

- Boeing 737-400 plane with 102 people on board crashed into a mountain in the West Sulawesi province of Indonesia, on Monday, January 01, 2007, killing at least 90 passengers, with 12 possible survivors.
- The plane was Adam Air flight KI-574, departing at 12:59 pm from Surabaya on Java bound for Manado in northeast Sulawesi.
- The plane crashed in a mountainous region in Polewali, west Sulawesi province.
- There were three Americans on board, it is not know if they survived.
- The cause of the crash is not known at this time but it is possible bad weather was a factor.

PG-Original Summary

- Adam Air Boeing 737-400 crashed Monday after vanishing off air traffic control radar screens between the Indonesian islands of Java and Sulawesi.
- Up to 12 people were thought to have survived, with rescue teams racing to the crash site near Polewali in West Sulawesi , some 180 kilometres north of the South Sulawesi provincial capital Makassar.
- It was the worst air disaster since Sept. 5, 2005, when a Mandala Airline's Boeing 737-200 crashed shortly after taking off from the North Sumatra's airport, killing 103 people.
- Earlier on Friday, a ferry carrying 628 people sank off the Java coast.

Extract+Rewrite Summary

- Plane with 102 people on board crashes.
- Three Americans among 102 on board plane in Indonesia.
- Rescue team arrives in Indonesia after plane crash.
- Plane with 102 crashes in West Sulawesi, killing at least 90.
- No word on the fate of Boeing 737-400.
- Plane carrying 96 passengers loses contact with Makassar.
- Plane crashes in Indonesia , killing at least 90.
- Indonesian navy sends two planes to carry bodies of five.
- Indonesian plane carrying 102 missing.
- Indonesian lawmaker criticises slow deployment of plane.
- Hundreds of kilometers plane crash.

PG-MMR Summary

- The Adam Air Boeing 737-400 crashed Monday afternoon, but search and rescue teams only discovered the wreckage early Tuesday.
- The Indonesian rescue team arrived at the mountainous area in West Sulawesi province where a passenger plane with 102 people onboard crashed into a mountain in Polewali, West Sulawesi province.
- Air force rear commander Eddy Suyanto told-Shinta radio station that the plane – operated by local carrier Adam Air – had crashed in a mountainous region in Polewali province on Monday.
- There was no word on the fate of the remaining 12 people on board the boeing 737-400.

Table 6: Example system summaries and human-written abstract. The sentences are manually de-tokenized for readability.

mary sentences do not appear in the source, but about 90% the n-grams do. The Extract+Rewrite summaries (§5), generated by rewriting selected source sentences to title-like summary sentences, exhibits a high degree of abstraction, close to that of human abstracts.

Linguistic quality. To assess the linguistic quality of various system summaries, we employ Amazon Mechanical Turk human evaluators to judge the summary quality, including PG-MMR, LexRank, PG-Original, and Extract+Rewrite. A turker is asked to rate each system summary on a scale of 1 (worst) to 5 (best) based on three evaluation criteria: ***informativeness*** (to what extent is the meaning expressed in the ground-truth text preserved in the summary?), ***fluency*** (is the summary grammatical and well-formed?), and ***non-redundancy*** (does the summary successfully avoid repeating information?). Human summaries are used as the ground-truth. The turkers are also asked to provide an overall ranking for the four system summaries. Results are presented in Table 5. We observe that the LexRank summaries are highest-rated on fluency. This is because LexRank is an extractive approach, where summary sentences are directly taken from the input. PG-MMR is rated as the best on both informativeness and non-redundancy. Regarding overall system rankings, PG-MMR summaries are frequently ranked as the 1st- and 2nd-best summaries, outperforming the others.

Example summaries. In Table 6 we present example summaries generated by various systems. PG-Original cannot effectively identify important content from the multi-document input. Extract+Rewrite tends to generate short, title-like sentences that are less informative and carry substantial redundancy. This is because the system is trained on the Gigaword dataset (Rush et al., 2015) where the target summary length is 7 words. PG-MMR generates summaries that effectively condense the important source content.

7 Conclusion

We describe a novel adaptation method to generate abstractive summaries from multi-document inputs. The method combines a strong extractive method (MMR) for sentence extraction and a recent abstractive model (PG) for fusing source sentences. The PG-MMR system demonstrates competitive results, substantially outperforming extractive and abstractive baselines.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J. Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. In *Proceedings of ACL*.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2017. Improving multi-document summarization via text classification. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Giuseppe Carenini and Jackie Chi Kit Cheung. 2008. Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality. In *Proceedings of the Fifth International Natural Language Generation Conference (INLG)*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based neural networks for document summarization. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of ACL*.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 update summarization task. In *Proceedings of Text Analysis Conference (TAC)*.

Hal Daume III and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Günes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*.

Giuseppe Di Fabbrizio, Amanda J. Stent, and Robert Gaizauskas. 2014. A hybrid approach to multi-document summarization of opinions in reviews. *Proceedings of the 8th International Natural Language Generation Conference (INLG)*.

Katja Filippova, Enrique Alfonseca, Carlos Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In *Proceedings of NAACL-HLT*.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the NAACL Workshop on Integer Linear Programming for Natural Langauge Processing*.

Dan Gillick, Benoit Favre, Dilek Hakkani-Tur, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009. The ICSI/UTD summarization system at TAC 2009. In *Proceedings of TAC*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of ACL*.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of Neural Information Processing Systems (NIPS)*.

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

Kai Hong, John M Conroy, Benoit 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)*.

Masaru Isonuma, Toru Fujino, Junichiro Mori, Yutaka Matsuo, and Ichiro Sakata. 2017. Extractive summarization using multi-task learning with document classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hongyan Jing and Kathleen McKeown. 1999. The decomposition of human-written summary sentences. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of EMNLP*.

Chin-Yew Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *Proceedings of ACL Workshop on Text Summarization Branches Out*.

Wencan Luo and Diane Litman. 2015. Summarizing student responses to reflection prompts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yishu Miao and Phil Blunsom. 2016. Language as a latent variable: Discrete generative models for sentence compression. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, 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)*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*.

Paul Over and James Yen. 2004. An introduction to DUC-2004. *National Institute of Standards and Technology*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Daniele Pighin, Marco Cornolti, Enrique Alfonseca, and Katja Filippova. 2014. Modelling events through memory-based, open-ie patterns for abstractive summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for sentence summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive document summarization with a graph-based attentional neural model. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alex Kuleszaand Ben Taskar. 2012. *Determinantal Point Processes for Machine Learning*. Now Publishers Inc.

Kapil Thadani and Kathleen McKeown. 2013. Sentence compression with joint structural inference. In *Proceedings of CoNLL*.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing and Management*, 43(6):1606–1618.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *Proceedings of ACL*.

Kam-Fai Wong, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*.

Dani Yogatama, Fei Liu, and Noah A. Smith. 2015. Extractive summarization by maximizing semantic volume. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP)*.

David Zajic, Bonnie J. Dorr, Jimmy Lin, and Richard Schwartz. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing and Management*.

Wenyuan Zeng, Wenjie Luo, Sanja Fidler, and Raquel Urtasun. 2017. Efficient summarization with read-again and copy mechanism. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.