CDEvalSumm: An Empirical Study of *C*ross-*D*ataset *Eval*uation for Neural *Summ*arization Systems

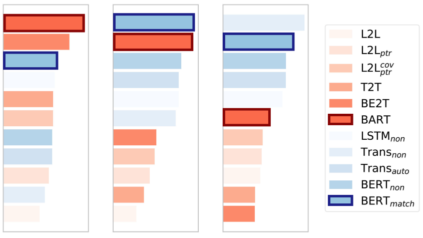
Yiran Chen[*]**, Pengfei Liu**[♯][*]**, Ming Zhong, Zi-Yi Dou**[♯]**, Danqing Wang,**
Xipeng Qiu[†]**, Xuanjing Huang**
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
2005 Songhu Road, Shanghai, China
♯Carnegie Mellon University
{yrchen19,mzhong18,dqwang18,xpqiu,xjhuang}@fudan.edu.cn
{zdou,pliu3}@cs.cmu.edu

Abstract

Neural network-based models augmented with unsupervised pre-trained knowledge have achieved impressive performance on text summarization. However, most existing evaluation methods are limited to an *in-domain* setting, where summarizers are trained and evaluated on the same dataset. We argue that this approach can narrow our understanding of the generalization ability for different summarization systems. In this paper, we perform an in-depth analysis of characteristics of different datasets and investigate the performance of different summarization models under a cross-dataset setting, in which a summarizer trained on one corpus will be evaluated on a range of out-of-domain corpora. A comprehensive study of 11 representative summarization systems on 5 datasets from different domains reveals the effect of model architectures and generation ways (i.e. abstractive and extractive) on model generalization ability. Further, experimental results shed light on the limitations of existing summarizers. Brief introduction and supplementary code can be found in https://github.com/zide05/CDEvalSumm.

1 Introduction

Neural summarizers have achieved impressive performance when evaluated by ROUGE (Lin, 2004) on in-domain setting, and the recent success of pre-trained models drives the state-of-the-art results on benchmarks to a new level (Liu and Lapata, 2019; Liu, 2019; Zhong et al., 2019a; Zhang et al., 2019; Lewis et al., 2019; Zhong et al., 2020). However, the superior performance is not a guarantee of a perfect system since exsiting models tend to show defects when evaluated from other aspects. For example, Zhang et al. (2018) observes that many abstractive systems tend to be near-extractive in practice. Cao et al. (2018); Wang et al. (2020); Kryściński et al. (2019); Maynez et al. (2020) reveal that most generated summaries are factually incorrect. These non-mainstream evaluation methods make it easier to identify the model's weaknesses.



(a) In-dataset R2 (b) Stiff-R2 (c) Stable-R2

Figure 1: Ranking (descending order) of current 11 top-scoring summarization systems (Abstractive models are red while extractive ones are blue). Each system is evaluated based on three diverse evaluation methods: (a) averaging each system's in-dataset ROUGE-2 F1 scores (R2) over five datasets; (b-c) evaluating systems using our designed cross-dataset measures: *stiff-R2*, *stable-R2* (Sec. 5). Notably, $BERT_{match}$ and *BART* are two state-of-the-art models for extractive and abstractive summarization respectively (highlighted by blue and red boxes).

Orthogonal to above two evaluation aspects, we aim to diagnose the limitation of existing systems under *cross-dataset evaluation*, in which a summarization system trained on one corpus would be evaluated on a range of out-of-dataset corpora. Instead of evaluating the quality of summarizers solely based on one dataset or multiple datasets individually, cross-dataset evaluation enables us to evaluate model performance from a different angle. For example, Fig. 1 shows the ranking of 11 summarization systems studied in this paper under different evaluation metrics, in which the ranking list "(a) in-dataset R2" is obtained by traditional ranking criteria while other two are based on our

[*] These two authors contributed equally.
[†] Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3679–3691
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

designed cross-dataset measures. Intuitively, we observe that 1) there are different definitions of a “good” system in various evaluation aspects; 2) abstractive and extractive systems exhibit diverse behaviors when evaluated under the cross-dataset setting.

The above example recaps the general motivation of this work, encouraging us to rethink the generalization ability of current top-scoring summarization systems from the perspective of cross-dataset evaluation. Specifically, we ask two questions as follows:

Q1: How do different neural architectures of summarizers influence the cross-dataset generalization performances? When designing summarization systems, a plethora of neural components can be adopted (Zhou et al., 2018; Chen and Bansal, 2018; Gehrmann et al., 2018; Cheng and Lapata, 2016; Nallapati et al., 2017). For example, will *copy* (Gu et al., 2016) and *coverage* (See et al., 2017) mechanisms improve the cross-dataset generalization ability of summarizers? Is there a risk that *BERT-based* summarizers will perform worse when adapted to new areas compared with the ones *without BERT*? So far, the generalization ability of current summarization systems when transferring to new datasets still remains unclear, which poses a significant challenge to design a reliable system in realistic scenarios. Thus, in this work, we take a closer look at the effect of model architectures on cross-dataset generalization setting.

Q2: Do different generation ways (*extractive* and *abstractive*) of summarizers influence the cross-dataset generalization ability? Extractive and abstractive models, as two typical ways to summarize texts, usually follow diverse learning frameworks and favor different datasets. It would be absorbing to know their discrepancy from the perspective of cross-dataset generalization. (e.g., whether abstractive summarizers are better at generating informative or faithful summaries on a new test set?)

To answer the questions above, we have conducted a comprehensive experimental analysis, which involves *eleven* summarization systems (including the state-of-the-art models), *five* benchmark datasets from different domains, and two evaluation aspects. Tab. 1 illustrates the overall analysis framework. We explore the effect of different architectures and generation ways on model generalization ability in order to answer *Q1* and *Q2*. Semantic equivalency (e.g., ROUGE) and factuality are adopted to characterize the different aspects of cross-dataset generalization ability. Additionally, we strengthen our analysis by presenting two views of evaluation: *holistic* and *fine-grained* views (Sec. 5).

Framework	**Semantic equivalency** (e.g., ROUGE)	**Factuality** (e.g., Factcc)
Q1: Architecture (e.g., Transformer v.s. LSTM)	Sec. 6.1.1	Sec. 6.2
Q2: Generation way (e.g., BERT v.s. BART)	Sec. 6.1.2	Sec. 6.2

Table 1: Overall analysis framework.

Our contributions can be summarized as: 1) Cross-dataset evaluation is orthogonal to other evaluation aspects (e.g., semantic equivalence, factuality), which can be used to re-evaluate current summarization systems, accelerating the creation of more robust summarization systems. 2) We have design two measures *Stiffness* and *Stableness*, which could help us to characterize generalization ability in different views, encouraging us to diagnose the weaknesses of state-of-the-art systems. 3) We conduct dataset bias-aided analysis (Sec. 4.3) and suggest that a better understanding of datasets will be helpful for us to interpret systems’ behaviours.

2 Representative Systems

Although it’s intractable to cover all neural summarization systems, we try to include more representative models to make a comprehensive evaluation. Our selection strategy follows: 1) the source codes of systems are publicly available; 2) systems with state-of-the-art performance or the top performace on benchmark datasets (e.g., CNNDM (Nallapati et al., 2016)) 3) systems equipped with typical neural components (e.g., Transformer, LSTM) or mechanism (e.g., copy).

2.1 Extractive Summarizers

Extractive summarizers directly choose and output the salient sentences (or phrases) in the original document. Generally, most of the existing extractive summarization systems follow a framework consisting of three major modules: *sentence encoder*, *document encoder* and *decoder*. In this paper, we investigate extractive summarizers with different choices of encoders and decoders.

LSTM$_{non}$ (Kedzie et al., 2018) This summarizer adopts convolutional neural network as sentence encoder and LSTM to model the cross-sentence relation. Finally, each sentence will be selected in a non-autoregressive way.

Trans$_{non}$ (Liu and Lapata, 2019) The TransformerExt model in Liu and Lapata (2019), similar to above setting except that the document encoder is replaced with the Transformer layer.

Trans$_{auto}$ (Zhong et al., 2019a) The decoder is replaced with a pointer network to avoid the repetition (autoregressive).

BERT$_{non}$ (Liu and Lapata, 2019) The BertSumExt model in Liu and Lapata (2019), this model is an extension of Trans$_{non}$ by introducing a BERT (Devlin et al., 2018) layer.

BERT$_{match}$ (Zhong et al., 2020) This is the existing state-of-the-art extractive summarization system, which introduce a matching layer using siamese BERT.

2.2 Abstractive Summarizers

The abstractive approach involves paraphrasing the inputs using novel words. The current abstractive summarization systems mainly focus on the *encoder-decoder* paradigm.

L2L$^{cov}_{ptr}$ (See et al., 2017) The model is a LSTM based sequence to sequence summarizer with copy and coverage mechanism.

L2L$_{ptr}$ We remove the coverage module and keep other parts unchanged.

L2L This model is implemented by removing the pointer network of the above summarizer.

T2T (Liu and Lapata, 2019) A sequence to sequence model with Transformer as the encoder and decoder.

BE2T (Liu and Lapata, 2019) A sequence to sequence model with BERT as encoder and Transformer as decoder.

BART (Lewis et al., 2019) A fully pre-trained sequence to sequence model. It is the existing state-of-the-art abstractive summarization system.

3 Datasets

We explore five typical summarization datasets: `CNNDM`, `Xsum`, `PubMed`, `Bigpatent B` and `Reddit TIFU`. `CNNDM` (Nallapati et al., 2016) and `Xsum` (Narayan et al., 2018) are news domain summarization datasets which are various in their publications and abstractiveness. `PubMed` (Cohan et al., 2018) is a scientific paper dataset, which can be used to investigate the generalization ability of models on scientific domain. `Bigpatent B` (Sharma et al., 2019) is the B category of `Bigpatent` (a dataset consisting of patent documents from Google Patents Public Datasets). `Reddit TIFU` (Kim et al., 2019) is a dataset with less formal posts collected from the online discussion forum Reddit. Detailed statistics and introduction of datasets are presented in the appendix section.

4 Evaluation for Summarization

Existing summarization systems are usually evaluated on different datasets individually based on an automatic metric: $r = \text{eval}(D, S, m)$, where D, S represents a dataset (e.g., `CNNDM`) and system (e.g., `L2L`) respectively. m denotes an evaluation metric (e.g., ROUGE).

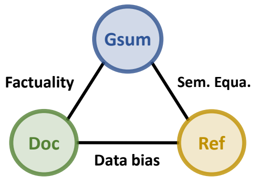


Figure 2: Different metrics characterized by a relation chart among generated summaries (`Gsum`), references (`Ref`) and input documents (`Doc`).

To evaluate the quality of generated summaries, metrics can be designed from diverse perspectives, which can be abstractly characterized in Fig. 2. Specifically, *semantic equivalence* is used to quantify the relation between generated summaries (`Gsum`) and references (`Ref`) while *factuality* aims to characterize the relation between generated summaries (`Gsum`) and input documents (`Doc`).

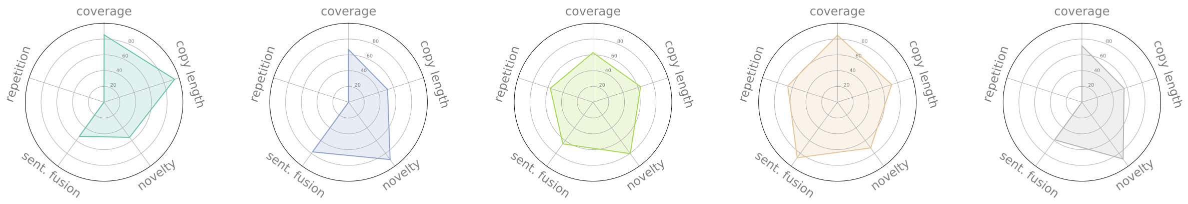
Besides evaluation metrics, in this paper, we also introduce some measures that quantify the relation between input documents (`Doc`) and references (`Ref`). We claim that a better understanding of dataset biases can help us interpret models' discrepancies.

4.1 Semantic Equivalence

ROUGE (Lin, 2004) is a classic metric to evaluate the quality of model generated summaries by counting the number of overlapped n-grams between the evaluated summaries and the ideal references.

4.2 Factuality

Apart from evaluating the semantic equivalence between generated summaries and the references, another evaluation aspect of recent interest is *factuality*. In order to analyze the generalization performance of models in different perspectives, in this



(a) CNN. (b) Xsum (c) PubMed (d) Bigatent b (e) Reddit

Figure 3: Characteristics of test set for each dataset (the train set possesses almost the same property thus is not displayed here): *coverage*,*copy length*, *novelty*, *sentence fusion score*, *repetition*. Here we choose 2-gram to calculate the novelty and 3-gram for the repetition.

work, we also take the factuality evaluation into consideration.

Factcc *Factcc* (Kryściński et al., 2019) is introduced to measure the fact consistency between the generated summaries and source documents. It is a model based metric which is weakly-supervised. We use the proportion of summary sentences that factcc predicts as factually consistent as the factuality score in this paper.

4.3 Dataset Bias

We detail several measures that could quantify the characteristics of datasets, which are helpful for us to understand the differences among models.

Coverage (Grusky et al., 2018) illustrates the overlap rate between document and summary, it is defined as the proportion of the copied segments in summary.

Copy Length measures the average length of segments in summary copied from source document.

Novelty (See et al., 2017) is defined as the proportion of segments in the summaries that haven't appeared in source documents. The segments can be instantiated as n-grams.

Repetition (See et al., 2017) measures the rate of repeated segments in summaries. Similar to the above measure, we choose n-gram (n ranges from one to four) as segment unit.

Sentence fusion score is calculated using the result of the algorithm proposed by (Lebanoff et al., 2019), which is to find whether summary sentence is compressed from one sentence or fused from several sentences. Then, sentence fusion score is calculated as the proportion of *fused sentences* (sentences that are fused from two or three document sentences) to all summary sentences.

A high value of coverage and copy length suggests the dataset is more extractive, while novelty represents the rate of novel units in summary and sentence fusion score represents the proportion of sentences that is fused from more than two document sentences. Zhong et al. (2019b) also explores dataset bias to aid the analysis of model performance, but they only focus on metrics for extractive summarizers.

4.4 Dataset Bias Analysis

According to the coverage and copy length results in Fig. 3, `CNNDM` is the most extractive dataset. `Bigpatent B` also exhibits relatively higher copy rate in summary but the copy segments is shorter than `CNNDM`. On the other hand, `Bigaptent b`, `Xsum` obtain higher sentence fusion score, which suggests that the proportion of *fused sentences* in these two datasets are high. `Xsum` and `Reddit` obtain more 3-gram novel units in summary, reflecting these two datasets are more abstractive. In terms of repetition in Fig. 3, only `PubMed` and `Bigpatent B` contain more 2-gram repeated phrases in summary.

	Models	**ROUGE 1**					
		CNN.*	**CNN.**	**Xsum**	**Pubm.**	**Patent b**	**Red.**
Ext.	LSTM$_{non}$	41.22	41.36	19.51	42.98	39.29	20.46
	Trans$_{non}$	40.90	40.84	15.74	38.45	34.41	16.25
	Trans$_{auto}$	41.36	41.35	19.29	42.74	38.76	18.55
	BERT$_{non}$	43.25	42.69	21.76	38.74	35.85	21.84
	BERT$_{match}$	44.22	44.26	24.97	41.19	38.89	25.32
Abs.	L2L	31.33	32.80	28.31	27.84	30.46	16.89
	L2L$_{ptr}$	36.44	37.06	29.67	32.04	31.03	21.32
	L2L$_{ptr}^{cov}$	39.53	39.95	28.83	35.27	35.90	21.28
	T2T	40.21	39.90	29.01	30.71	42.94	19.96
	BE2T	41.72	41.34	38.99	37.11	43.10	26.66
	BART	44.16	44.75	44.73	45.02	45.78	34.00

Table 2: Representative summarizers studied in this paper and their corresponding performance (ROUGE-1 F1 score) on different datasets (`CNNDM`, `Xsum`, `PubMed`, `Bigpatent B`, `Reddit`). We reimplement all 11 systems on five datasets by ourselves. All implemented results can outperform or slightly lower than the performances reported in original papers (the column of CNN.*).

	$\mathbf{U}_A$			$\mathbf{U}_B$			Measures	
	a	b		a	b		$\mathbf{U}_A$	$\mathbf{U}_B$
a	48	40	a	61	43	*Stiff.*	44	55
b	41	45	b	46	69	*Stable.*	94	84

Table 3: Illustration of two views (*Stiffness*: r^u and *Stableness*: r^σ) to characterize the cross-dataset (a and b) generalization based on model A and B. $\mathbf{U_A}$ and $\mathbf{U_B}$ represent two cross-dataset matrix of two models. $r^\mu(\mathbf{U_A}) < r^\mu(\mathbf{U_B})$ means the model B gains a better cross-dataset absolute performance while $r^\sigma(\mathbf{U_A}) > r^\sigma(\mathbf{U_B})$ suggests the model A is more robust.

5 Cross-dataset Evaluation

Despite recent impressive results on diverse summarization datasets, modern summarization systems mainly focus on extensive in-dataset architecture engineering while ignore the generalization ability which is indispensable when systems are required to process samples from new datasets or domains. Therefore, instead of evaluating the quality of summarization system solely based on one dataset, we introduce cross-dataset evaluation (a summarizer (e.g., *L2L*) trained on one dataset (e.g., `CNNDM`) will be evaluated on a range of other datasets (e.g., `XSUM`)). Methodologically, we perform cross-dataset evaluation from two views: fine-grained and holistic and we will detail them below.

5.1 Methodology

Given a summarization system S, a set of datasets $\mathcal{D} = D_1, \cdots, D_N$, and evaluation metric m, we can design different evaluation function to quantify the system's quality: $\mathbf{r} = \text{eval}(\mathcal{D}, S, m)$. Depending on different forms of function $\text{eval}(\cdot)$, $\mathbf{r}$ could be instantiated as either a scalar or a vector (or matrix).

5.1.1 Fine-grained Measures

Once $\mathbf{r}$, the cross-dataset evaluation result, is instantiated as a matrix, we can characterize the given system in a fine-grained way. Specifically, we define $\mathbf{r}$ as: $\mathbf{r} = \mathbf{U} \in \mathbb{R}^{N \times N}$ where each cell $\mathbf{U}_{i,j}$ refers to the metric result (e.g., ROUGE) when a summarizer is trained in dataset D_i and tested in dataset D_j (N refers to the number of datasets).

Additionally, we can normalize each cell by the diagonal value, $\mathbf{r} = \mathbf{U}_{ij}/\mathbf{U}_{jj} \times 100\% = \hat{\mathbf{U}}$, $\mathbf{U}_{ij}/\mathbf{U}_{jj}$ measures how close the out-of-dataset performance (trained in D_i and tested in D_j) of a system is to its in-dataset performance (trained in D_j and tested in D_j).

5.1.2 Holistic Measures

Instead of using a matrix, holistically, we can quantify the cross-dataset generalization ability of each summarization system using a scalar. Specifically, we propose two views to characterize the cross-dataset generalization.

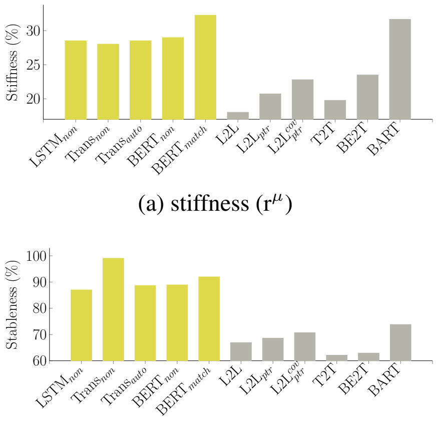
Stiffness This measure reflects the absolute performance of a system under cross-dataset setting. Given a system, its *stiffness* can be calculated as: $r^\mu = \frac{1}{N \times N} \sum_{i,j} \mathbf{U}_{ij}$

Intuitively, a higher value of *stiffness* suggests the system obtains better performance when transferred to new datasets.

Stableness It characterizes the relative performance gap between in-dataset and cross-dataset test. $r^\sigma = \frac{1}{N \times N} \sum_{i,j} \mathbf{U}_{ij}/\mathbf{U}_{jj} \times 100\%$

Generally, a higher value of *stableness* suggests that the variance between in-dataset and cross-dataset results is smaller.

Tab. 3 gives an example to characterize generalization ability in two views. It shows that stiffness and stableness are not always unanimous, a model with higher stiffness may obtains lower stableness.



(a) stiffness (r^μ)

(b) stableness (r^σ)

Figure 4: Illustration of stiffness and stableness of ROUGE-1 F1 scores for various models. Yellow bars stand for extractive models and grey bars stand for abstractive models.

6 Experiment

In what follows, we analyze different summarization systems in terms of semantic equivalence and factuality. Moreover, the results are studied in holistic and fine-grained views based on the measures defined above. Holistic results are showed in Fig. 4

analysis aspect			Architecture											
model type			EXT											
compare models			BERT$_{match}$ vs. BERT$_{non}$						BERT$_{non}$ vs. Trans$_{non}$					
holistic analysis			stiff. : 32.27 vs. 28.98 stable. : 91.98 vs. 88.93						stiff. : 28.98 vs. 28.02 stable. : 88.93 vs. 99.05					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg	CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	1.6	4.1	4.5	3.0	4.7	3.6	1.8	1.2	0.3	0.8	-10.9	-1.3
		Xsum	2.9	3.2	3.5	1.6	5.7	3.4	-0.9	6.0	0.1	-1.6	-0.7	0.6
		Pubm.	0.9	4.0	2.4	0.2	8.7	3.3	2.5	1.4	0.3	0.6	-2.2	0.5
		Patent b	4.6	3.1	3.5	3.0	3.7	3.6	0.5	1.1	0.2	1.4	3.8	1.4
		Red.	3.3	4.2	3.5	-1.4	3.5	2.6	8.3	3.0	-0.1	1.6	5.6	3.7
		avg	2.6	3.7	3.5	1.3	5.3	3.3	2.4	2.5	0.2	0.6	-0.9	1.0
			(a)						(b)					
	normali.	CNN.	0.0	5.8	5.3	0.7	6.9	3.7	0.0	-23.9	0.1	-1.5	-96.6	-24.4
		Xsum	3.4	0.0	2.8	-2.7	11.5	3.0	-6.1	0.0	-0.5	-8.3	-31.8	-9.3
		Pubm.	-1.2	6.1	0.0	-6.5	26.5	5.0	2.0	-21.0	0.0	-2.2	-33.7	-11.0
		Patent b	7.3	1.8	2.8	0.0	3.3	3.0	-2.6	-24.8	-0.2	0.0	-5.5	-6.6
		Red.	4.4	6.2	2.9	-10.5	0.0	0.6	16.3	-12.8	-1.0	1.0	0.0	0.7
		avg	2.8	4.0	2.7	-3.8	9.6	3.1	1.9	-16.5	-0.3	-2.2	-33.5	-10.1
			(g)						(h)					

analysis aspect			Architecture											
model type			ABS											
compare models			L2L$_{ptr}$ vs. L2L						L2L$_{ptr}^{cov}$ vs. L2L$_{ptr}$					
holistic analysis			stiff. : 20.74 vs. 18.03 stable. : 68.63 vs. 66.93						stiff. : 22.81 vs. 20.74 stable. : 70.71 vs. 68.63					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg	CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	4.3	0.5	5.3	3.2	1.5	3.0	2.9	1.8	6.4	3.4	1.7	3.2
		Xsum	3.4	1.4	3.4	4.2	0.1	2.5	-0.8	-0.8	-4.5	-2.4	-0.1	-1.7
		Pubm.	10.3	2.3	4.2	3.0	2.6	4.5	4.5	1.7	3.2	3.4	2.7	3.1
		Patent b	1.1	-1.1	2.5	0.6	-0.3	0.5	1.0	2.0	2.2	4.9	0.8	2.2
		Red.	2.2	3.1	2.6	2.9	4.4	3.0	3.3	1.0	6.5	6.9	-0.0	3.5
		avg	4.2	1.2	3.6	2.8	1.7	2.7	2.2	1.1	2.8	3.2	1.0	2.1
			(c)						(d)					
	normali.	CNN.	0.0	-1.0	4.8	8.7	-9.9	0.5	0.0	8.1	9.6	-4.1	8.0	4.3
		Xsum	1.8	0.0	0.7	12.2	-13.8	0.2	-6.7	0.0	-19.7	-18.0	-0.2	-8.9
		Pubm.	23.3	5.6	0.0	8.4	1.6	7.8	6.7	7.4	0.0	-1.2	12.6	5.1
		Patent b	-1.6	-5.8	-0.4	0.0	-14.5	-4.4	-0.1	8.1	0.8	0.0	3.7	2.5
		Red.	1.9	8.7	3.4	8.4	0.0	4.5	5.6	4.9	14.8	11.7	0.0	7.4
		avg	5.1	1.5	1.7	7.5	-7.3	1.7	1.1	5.7	1.1	-2.3	4.8	2.1
			(i)						(j)					

analysis aspect			Generation way											
model type			LSTM						BERTSUM					
compare models			LSTM$_{non}$ vs. L2L						BERT$_{non}$ vs. BE2T					
holistic analysis			stiff. : 28.51 vs. 18.03 stable. : 87.00 vs. 66.93						stiff. : 28.98 vs. 23.49 stable. : 88.93 vs. 62.93					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg	CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	8.6	0.1	13.2	4.9	2.0	5.7	1.3	-2.0	3.5	-1.8	-1.7	-0.1
		Xsum	13.1	-8.8	18.3	7.1	3.8	6.7	12.9	-17.2	18.3	9.9	1.5	5.1
		Pubm.	18.6	4.8	15.1	11.1	9.0	11.7	17.2	2.9	1.6	-0.3	0.3	4.3
		Patent b	19.7	2.8	22.8	8.8	5.9	12.0	21.8	6.7	15.4	-7.2	5.1	8.4
		Red.	21.4	7.3	30.7	18.0	3.6	16.2	17.8	4.6	20.2	11.4	-4.8	9.8
		avg	16.3	1.2	20.0	10.0	4.9	10.5	14.2	-1.0	11.8	2.4	0.1	5.5
			(e)						(f)					
	normali.	CNN.	0.0	28.4	-0.7	-7.9	-4.8	3.0	0.0	31.5	5.2	11.1	9.0	11.4
		Xsum	18.3	0.0	15.8	2.0	6.6	8.5	28.4	0.0	45.0	37.8	19.9	26.2
		Pubm.	36.8	44.3	0.0	12.5	35.1	25.7	38.7	42.0	0.0	14.5	11.5	21.3
		Patent b	39.6	35.2	31.4	0.0	17.8	24.8	49.9	53.7	37.2	0.0	34.1	35.0
		Red.	44.7	52.9	58.4	35.1	0.0	38.2	40.1	48.4	50.2	41.5	0.0	36.1
		avg	27.9	32.2	21.0	8.3	10.9	20.1	31.4	35.1	27.5	21.0	14.9	26.0
			(k)						(l)					

Table 4: The difference of ROUGE-1 F1 scores between different model pairs. Every column of the table represents the compared results of one pair of models. The line of holistic analysis displays the overall stiffness and stableness of compared models. The rest of the table is fine-grained results, the first line of which is the origin compared results ($\mathbf{U_A} - \mathbf{U_B}$ for model pairs A and B) and the second line is the normalized compared results ($\mathbf{\hat{U}_A} - \mathbf{\hat{U}_B}$ for model pairs A and B). For all heatmap, 'grey' and 'red' represent positive and negative respectively. Here we only display compared results for limited pairs of models, all other results are displayed in appendix.

and Fig. 5. On the other hand, Tab. 4 and Tab. 5 display the fine-grained observations. Tab. 2 dispalys the in-dataset results of all models on five benchmark datasets.

6.1 Semantic Equivalence Analysis

We conduct pair-wise Wilcoxon Signed-Rank significant test with $\alpha = 0.05$. The null hypothesis is that the expected performances (stiffness and stableness) of a pair of summarization models are identical. We report the observations that are statistically significant.

6.1.1 Architecture

Match based reranking improves stiffness significantly *BERT$_{match}$*, which using semantic match scores to rerank candidate summaries enhances the stiffness of model significantly in Fig. 4a while obtaining comparable stableness with other extractive models in Fig. 4b. This indicates that *BERT$_{match}$* not only increases the absolute performance but also retaining robustness.

***BERT$_{match}$* is not stable when transferred from other datasets to `Bigpatent B`** As Tab. 4g shows, when compared to *BERT$_{non}$*, *BERT$_{match}$* obtains larger in-dataset and cross-dataset performance gap when tested in `Bigpatent B`. This is because `Bigpatent B` possesses higher sentence fusion score and higher repetition compared with other datasets as Sec. 4.4 demonstrates. When served as test set, such dataset brings great challenge for *BERT$_{match}$* to correctly rank the candidate summaries while it provides more training signals when served as training set. Thus the in-dataset (`Bigpatent b`) trained model obtain much higher score compared with cross-dataset models which trained from other datasets and cause lower stableness.

Non-autoregressive decoder is more robust than autoregressive for extractive models. Regarding the decoder of extractive systems, as shown in Fig. 4a and Fig. 4b, the non-autoregressive extractive decoder (*Trans$_{non}$*) is more stable while it possesses lower stiffness than its autoregressive counterpart (*Trans$_{auto}$*).

Pointer network and coverage mechanism are instrumental in improving stiffness and stableness of abstractive systems. The pointer network and coverage mechanism do enhance the absolute performance of abstractive system as Fig. 4a demonstrates ($\mathrm{r}^{\mu}(L2L^{cov}_{ptr}) > \mathrm{r}^{\mu}(L2L_{ptr}) > \mathrm{r}^{\mu}(L2L)$). Also, the stableness results of *L2L$_{ptr}$* and *L2L* in Fig. 4b reveals that once removing the pointer mechanism, the value of r^{σ} for *L2L$_{ptr}$* decreases, which suggests that *the system will be more stable if it's augmented the ability to directly extract text spans from the the source document*.

However, pointer network brings trivial improvement when tested in `Xsum` and `Reddit` The absolute model performance improvement of pointer network is trival when tested in `xsum` and `Reddit` as showed in Tab. 4c, which is in line with expectations because these two datasets are more abstractive as analyzed in Sec. 4.4.

On the other hand, coverage is not that helpful when tested in `Reddit` and `Xsum` and even harmful when trained in `Xsum`. The heatmap of $L2L_{ptr}^{cov}$ vs. $L2L_{ptr}$ in Tab.4d) shows that when tested in `Reddit` and `Xsum`, the improvement of coverage mechanism is trivial. These two datasets possess less repetition, thus coverage can not provide much help when transferred to these datasets. Moreover, when trained in `Xsum`, $L2L_{ptr}^{cov}$ gets lower stiffness compared with $L2L_{ptr}$, which is in accordance with the normalized result in Tab. 4j. This is because the gold summaries of `Xsum` exhibit lower repetition score (as analyzed in Sec. 4.4), thus can't provide enough learning signals for coverage mechanism.

BERT sometimes brings unstableness. As shown in Fig. 4a, there is no doubt that once summarizers (extractive or abstractive) are equipped with pre-trained encoder, the stiffness will increase significantly (e.g., $r^{\mu}(BE2T >> r^{\mu}(T2T)$, suggesting that the overall cross-dataset performance has been improved. *However, we are surprised to find (from Fig. 4b) that BERT sometimes leads to unstableness* (i.e. $r^{\sigma}(Trans_{non}) > r^{\sigma}(BERT_{non})$). This result enlightens us to search for other architectures or learning schemas to offset the unstableness brought by BERT.

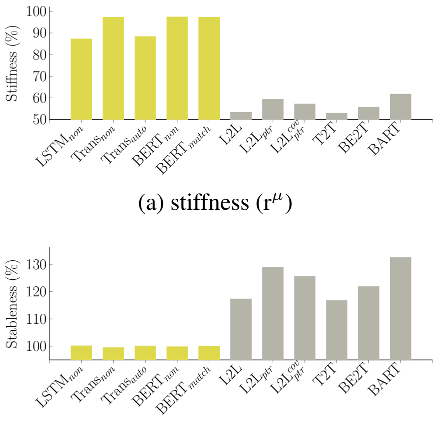
As the heatmap of $BERT_{non}$ vs. $Trans_{non}$ in Tab. 4h shows, BERT brings unstableness especially when tested in `Reddit` and `Xsum`.

BERT sometimes can even harm the absolute cross-dataset performance. $BERT_{non}$ performs worse than $Trans_{non}$ in some cells (e.g., trained in `Xsum` and tested in `CNNDM`) in Tab. 4b

BART shows superior performance in terms of stiffness and stableness. As Fig. 4a shows, *BART* obtains the highest stiffness among all abstractive models, and is even comparable with $BERT_{match}$. In addition, *BART* is also outstanding in terms of stableness when compared with other abstractive models (Fig. 4b). The performance gap between *BART* and *BE2T* proves that for abstractive models, pre-training the whole sequence to sequence model works better than using the pre-trained model in either side of encoder or decoder.

6.1.2 Generation ways

Extractive models are superior to abstractive models in terms of stiffness and robustness. Extractive models show superior advantage of absolute performance as shown in Fig. 4a. Moreover, comparing the stableness of abstractive and extractive models in Fig. 4b, we surprisingly find that *abstractive approaches except for BART are extremely brittle* since their r^{σ} value is much lower than any extractive approaches with a maximum margin of 37%, and the gap can be reduced by introducing pointer network. This observation poses a great challenge to the development of the abstractive systems, encouraging research to pay more attention to improve the generalization ability. Also, we have provided hints for the solution, such as enabling the model to extract granular information from the source document or using the well pre-trained sequence to sequence model (e.g., *BART*).



(a) stiffness (r^{μ})

(b) stableness (r^{σ})

Figure 5: Illustration of stiffness and stableness of factuality scores for various models. Yellow bars stand for extractive systems and grey bars stand for abstractive systems.

When tested in `Xsum` and `Reddit`, abstractive systems possess comparable or even better performance. The supremacy of extractive models is not retained in all datasets (Tab. 4f and Tab. 4e) Though extractive models obtain higher stiffness scores when tested in `CNNDM` and `PubMed`, abstractive approaches (*BE2T*, *L2L*) obtained higher or comparable stiffness scores when tested at `XSUM` and `Reddit`. This is because `Xsum` and `Reddit` are more abstractive as analyzed in Sec. 4.4.

6.2 Factuality Analysis

1) All extractive models can achieve higher factuality scores while all abstractive models obtain quite lower ones (Fig. 5a). One interesting observation is, for extractive models, not all factuality scores under the in-dataset setting are 100% in Tab. 5 (on-diagonal values), which reveals the limitation of

EXT models	Trans_{non}						BERT_{match}					
	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg
CNN	100.0	100.0	98.0	99.1	100.0	99.4	99.8	99.4	92.9	95.7	99.1	97.4
XSUM	99.8	100.0	97.4	98.2	100.0	99.1	99.7	99.5	93.2	95.1	98.8	97.3
Pubm.	97.7	98.8	95.1	94.7	100.0	97.3	99.7	99.2	93.1	95.2	99.3	97.3
Patent B	98.3	99.8	96.3	97.4	99.5	98.3	99.7	99.0	93.0	94.5	98.4	96.9
Reddit	90.3	94.1	94.1	86.7	96.3	92.3	99.7	99.3	93.1	96.1	99.3	97.5
avg	97.2	98.6	96.2	95.2	99.2	97.3	99.7	99.3	93.0	95.3	99.0	97.3

ABS models	T2T						BART					
	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg
CNN	72.4	75.7	71.5	71.8	70.5	72.4	69.9	77.9	87.4	84.1	90.2	81.9
XSUM	9.7	22.6	10.8	9.9	19.1	14.4	35.5	24.7	36.1	50.1	50.7	39.4
Pubm.	58.5	59.3	56.2	72.3	34.9	56.2	69.5	61.5	58.4	61.3	94.1	69.0
Patent B	79.2	81.2	84.4	68.7	73.9	77.5	52.1	53.8	69.0	67.4	76.8	63.8
Reddit	34.8	35.7	50.6	44.6	52.5	43.6	59.6	50.3	69.1	49.3	44.2	54.5
avg	50.9	54.9	54.7	53.5	50.2	52.8	57.3	53.6	64.0	62.4	71.2	61.7

Table 5: Cross-dataset factuality scores for extractive and abstractive models.

existing factuality checker.
2) *BART* can significantly improve the ability to generate factual summaries compared with other abstractive models as showed in Fig. 5a, even compared with $L2L_{ptr}$ which equipped with pointer network and tend to copy from source document.
3) Abstractive models obtain higher stableness of factuality scores in Fig. 5b which surpass 100%. This is because when tested in abstractive datasets (e.g., `Xsum` as Sec. 4.4 shows), abstractive summarizers trained in-dataset tend to be more abstractive and obtain lower factuality score while it gets higher factuality score when trained on other datasets which are more extractive (e.g., `CNNDM`). The superiority of cross-dataset results over in-dataset results thus leads to higher stableness.

7 Related Work

Our work is connected to the following threads of topics of NLP research.

Cross-Dataset Generalization in NLP Recently, more researchers shift their focus from individual dataset to cross-dataset evaluation, aiming to get a comprehensive understanding of system's generalization ability. Fried et al. (2019) explores the generalization ability of different constituency parsers. Talmor and Berant (2019), on the other hand, shows the generalization ability of reading comprehension models can be improved by pre-training on one or two other reading comprehension datasets. Fu et al. (2020) studies the model generalization in the field of NER. They point out the bottleneck of the existing NER systems through in-depth analyses and provide suggestions for further improvement. Different from the above works, we attempt to explore generalization ability for summarization systems.

Diagnosing Limitations of Existing Summarization Systems Beyond ROUGE, some recent works try to explore the weaknesses of existing systems from divese aspects. Zhang et al. (2018) tries to figure out to what extent the neural abstractive summarization systems are abstractive and discovers many of abstractive systems tend to perform near-extractive. On the other hand, Cao et al. (2018) and Kryściński et al. (2019) study the factuality problem in modern neural summarization systems. The former puts forward one model that combining source document and preliminary extracted fact description and prove the effectiveness of this model in terms of factuality correctness. While the latter contributes to design a model-based automatic factuality evaluation metric. Abstractiveness and factuality error the above works studied are orthogonal to this work and can be easily combined with cross-dataset evaluation framework in this paper as Sec. 6.2 shows. Moreover, Wang et al. (2019); Hua and Wang (2017) attempt to investigate the domain shift problem on text summarization while they focus on a single generation way (either abstractive or extractive) We also investigate the generalization of summarizers when transferring to different datasets, but include more datasets and models.

8 Conclusion

By performing a comprehensive evaluation on eleven summarization systems and five mainstream datasets, we summarize our observations below:

1) Abstractive summarizers are extremely brittle compared with extractive approaches, and the maximum gap between them reaches 37% in terms of the measure *stableness* (ROUGE) defined in this paper. 2) *BART* (SOTA system) is superior over other abstractive models and even comparable with extractive models in terms of stiffness (ROUGE). On the other hand, it is robust when transferring between datasets as it possesses high stableness (ROUGE). 3) $BERT_{match}$ (SOTA system) performs excellently in terms of stiffness, while still lacks stableness when transferred to `Bigpatent B` from other datasets. 4) The robustness of models can be improved through either equipped the model with ability to copy span from source document (i.e., Lebanoff et al. (2019)) or make use of well trained sequence to sequence pre-trained model (*BART*). 5) Simply adding BERT on encoder could improve the stiffness (ROUGE) of model but will cause larger cross-dataset and in-dataset perfor-

mance gap, a better way should be found to merge BERT into abstractive model, or a better training strategy should be applied to offset the negative influence it brings. 6) Existing factuality checker (Factcc) is limited in predictive power of positive samples (Sec.6.2). 7) Out-of-domain systems can even surpass in-domain systems in terms of factuality. (Sec.6.2)

Acknowledgements

We would like to thank the anonymous reviewers for their detailed comments and constructive suggestions. This work was supported by the National Natural Science Foundation of China (No. 62022027 and 61976056), Science and Technology on Parallel and Distributed Processing Laboratory (PDL).

References

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

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 (Volume 1: Long Papers)*, volume 1, pages 675–686.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 484–494.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, volume 2, pages 615–621.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Daniel Fried, Nikita Kitaev, and Dan Klein. 2019. Cross-domain generalization of neural constituency parsers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 323–330, Florence, Italy. Association for Computational Linguistics.

Jinlan Fu, Pengfei Liu, Qi Zhang, and Xuanjing Huang. 2020. Rethinking generalization of neural models: A named entity recognition case study. *arXiv preprint arXiv:2001.03844*.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 708–719.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1684–1692.

Xinyu Hua and Lu Wang. 2017. A pilot study of domain adaptation effect for neural abstractive summarization. *arXiv preprint arXiv:1707.07062*.

Chris Kedzie, Kathleen McKeown, and Hal Daume III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2519–2531.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv*, pages arXiv–1910.

Logan Lebanoff, Kaiqiang Song, Franck Dernoncourt, Doo Soon Kim, Seokhwan Kim, Walter Chang, and Fei Liu. 2019. Scoring sentence singletons and pairs for abstractive summarization. *arXiv preprint arXiv:1906.00077*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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

Yang Liu. 2019. Fine-tune BERT for Extractive Summarization.

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)*, pages 3721–3731.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Ça glar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *CoNLL 2016*, page 280.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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 (Volume 1: Long Papers)*, volume 1, pages 1073–1083.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. *arXiv preprint arXiv:1906.03741*.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. *arXiv preprint arXiv:2004.04228*.

Danqing Wang, Pengfei Liu, Ming Zhong, Jie Fu, Xipeng Qiu, and Xuanjing Huang. 2019. Exploring domain shift in extractive text summarization. *arXiv preprint arXiv:1908.11664*.

Fangfang Zhang, Jin-ge Yao, and Rui Yan. 2018. On the abstractiveness of neural document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 785–790, Brussels, Belgium. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *arXiv preprint arXiv:1912.08777*.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuan-Jing Huang. 2019a. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1049–1058.

Ming Zhong, Danqing Wang, Pengfei Liu, Xipeng Qiu, and Xuan-Jing Huang. 2019b. A closer look at data bias in neural extractive summarization models. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 80–89.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 654–663.

A Appendices

A.1 Detailed Dataset introduction

`CNN/DailyMail` The CNN/DailyMail question answering dataset (Hermann et al., 2015) modified by Nallapati et al. (2016) is commonly used for summarization. The dataset consists of online news articles with paired human-generated summaries. For the data preprocessing, we use the non-anonymized data as See et al. (2017), which doesn't replace named entities.

XSUM XSUM (Narayan et al., 2018) is a dataset consists of the articles and the single-sentence answers of the question "What is the article about?" as summary. It is more abstractive compared with CNN/DailyMail.

PUBMED PUBMED (Cohan et al., 2018) is drawn from scientific papers specifically medical journal articles from the PubMed Open Access Subset. We use the introduction as source document and the abstract as summary here.

BIGPATENT BIGPATENT (Sharma et al., 2019) consists of 1.3 million records of U.S. patent documents and the corresponding summaries are created by human. According to Cooperative Patent Classification (CPC), the dataset is divided to nine categories. One of the nine categories is chosen as a dataset in difference domain in our experiment (Category B: Performing Operations; Transporting).

REDDIT TIFU REDDIT TIFU (Kim et al., 2019) is a dataset with less formal posts compared with datasets mentioned above which mostly use formal documents as source. It is collected from the online discussion forum Reddit. They regard the body text as source, the title as short summary, and the TL;DR summary as long summary, thus making two sets of datasets: TIFU-short and TIFU-long. TIFU-long is used in this paper.

A.2 Dataset statistics

The detailed dataset statistics are presented in Tab. 6

Datasets	Statistics	Topics	Oracle	Lead-k
CNNDM	2,764/123/107M	News	55.21	40.32
Xsum	1126/60/59M	News	30.41	16.38
Pubmed	644/36/38M	Scientific	46.21	37.52
BigPatent B	4,812/265/262M	Patents	51.53	31.85
Reddit	206/3.3/3.6M	Posts	36.47	11.09

Table 6: Detailed statistics of five datasets. Lead-k indicates ROUGE-1 F1 score of the first k sentences in the document and Oracle indicates the globally optimal combination of sentences in terms of ROUGE-1 F1 scores with ground truth, the latter represents the upper bound of extractive models.

A.3 Experimental setup

A.3.1 Extractive Summarizers

We use the same training setup in (Zhong et al., 2019a). We use cross entropy as loss function to train $LSTM_{non}$ and $Trans_{auto}$. The hidden state dimension of LSTM in $LSTM_{non}$ is set to 512 and the hidden state dimension of Transformer in $Trans_{auto}$ is 2048. We use Transformer with 8 heads.

$BERT_{non}$ and $Trans_{non}$ is constructed according to Liu and Lapata (2019). All documents and summaries are truncated to 512 tokens when training. $BERT_{non}$ and $Trans_{non}$ are trained for 50000 steps, the gradient is accumulated every two steps. We use Adam as optimizer and the learning rate is set to 2e-3.

$BERT_{match}$ is trained as in Zhong et al. (2020). It uses the base version of BERT as base model. We use Adam optimizer with warming up. The learning rate schedule follows Vaswani et al. (2017).

A.3.2 Abstractive Summarizers

L2L, $L2L_{ptr}$ and $L2L_{ptr}^{cov}$ are trained using the pytorch reproduced version code of See et al. (2017). We use the same size of vocabulary(50k), hidden state dimension (256) and word embedding dimension (128) as in the paper. All of three models are trained with 650000 maximum training steps, We use Adagrad to train the models with learning rate of 0.15.

BE2T and *T2T* is constructed according to Liu and Lapata (2019). We use two separate optimizers for the decoder and encoder regarding *BE2T* to offset the mismatch of encoder and decoder, since the former is pre-trained while the latter is not. Learning rates for the optimizers of encoder and decoder are 0.002 and 0.2 respectively. On the other hand, *BE2T* and *T2T* are trained with gradient accumulation every five steps, training step for which is 200000.

BART uses the large pre-trained sequence to sequence model in Lewis et al. (2019). The total learning step when fine-tuning is set to 20000 with 500 steps warming up. We use Adam as optimizer and learning rate is 3e-05.

A.4 In-dataset ROUGE results for all models

Tab. 7 displays in-dataset ROUGE-1 F1 ,ROUGE-2 F1 ,ROUGE-L F1 scores.

A.5 The ROUGE-1 F1 score difference of all model pairs which are meaningful to compare

The holistic and fine-grained results of pair-wise comparison are displayed in Tab. 10.

	Models	CNNDM			XSUM			PubMed			Bigpatent b			Reddit		
		R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
Ext.	LSTM$_{non}$	41.36	18.81	37.73	19.51	3.10	14.50	42.98	16.59	38.28	39.29	13.07	32.61	20.46	5.05	16.33
	Trans$_{non}$	40.84	18.23	37.09	15.74	1.67	11.58	38.45	13.28	34.16	34.41	10.05	28.75	16.25	2.60	12.57
	Trans$_{auto}$	41.35	18.77	37.75	19.29	2.80	14.21	42.74	16.34	38.05	38.76	12.60	32.17	18.55	3.44	14.62
	BERT$_{non}$	42.69	19.88	38.99	21.76	4.24	16.00	38.74	13.62	34.48	35.85	11.05	29.97	21.84	5.21	17.15
	BERT$_{match}$	44.26	20.58	40.40	24.97	4.76	18.48	41.19	14.91	36.73	38.89	12.82	32.48	25.32	6.16	20.17
Abs.	L2L	32.80	12.84	30.34	28.31	8.71	22.30	27.84	7.45	25.69	30.46	9.76	27.61	16.89	1.24	13.63
	L2L$_{ptr}$	37.06	15.96	33.74	29.67	9.58	23.40	32.04	10.38	28.97	31.03	9.92	25.35	21.32	4.46	17.14
	L2L$_{ptr}^{cov}$	39.95	17.54	36.25	28.83	8.83	22.62	35.27	11.89	31.92	35.90	12.31	32.78	21.28	4.39	17.22
	T2T	39.90	17.66	37.08	29.01	9.13	22.77	30.71	8.10	27.97	42.94	16.75	37.06	19.96	3.36	15.60
	BE2T	41.34	18.98	38.41	38.99	16.64	31.23	37.11	13.38	33.72	43.10	17.11	37.34	26.66	7.00	21.21
	BART	44.75	21.69	41.46	44.73	21.99	37.02	45.02	16.94	41.17	45.78	18.31	38.98	34.00	11.88	26.91

Table 7: Representative summarizers we have studied in this paper and their correspond performance (ROUGE-1 F1, ROUGE-2 F1, ROUGE-L F1) on different datasets.

A.6 Cross-dataset factuality results of all models

The cross-dataset factcc results for abstractive models are shown in Tab. 8 and the factcc results of extractive models are demonstrated in Tab. 9.

A.7 Code urls

A.7.1 Training code urls

The models and their training code urls are listed below:

LSTM$_{non}$ and *Trans$_{auto}$* are trained from the code in Zhong et al. (2019a), the code url is https://github.com/maszhongming/Effective_Extractive_Summarization.

We use the code from Liu and Lapata (2019) for *BERT$_{non}$*, *Trans$_{non}$*, *BE2T* and *T2T*. Code url is https://github.com/nlpyang/PreSumm.

BERT$_{match}$ uses the code from Zhong et al. (2020) and the code url is https://github.com/maszhongming/MatchSum.

L2L, *L2L$_{ptr}$* and *L2L$_{ptr}^{cov}$* are trained from the code of See et al. (2017), code url is https://github.com/atulkum/pointer_summarizer.

We use code in fairseq (Ott et al., 2019) to fine-tune *BART*, the code url is https://github.com/pytorch/fairseq/tree/master/examples/bart.

A.7.2 Evaluation code urls

The evaluation metrics code urls are listed below:

We use pyrouge (https://github.com/bheinzerling/pyrouge) to evaluate the ROUGE performance of models.

The url for Factcc (Kryściński et al., 2019) is https://github.com/salesforce/factCC.

The url for other metrics for dataset bias is https://github.com/zide05/CDEvalSumm/tree/master/Data-bias-metrics.

ABS models	L2L						L2L$_{ptr}$						L2L$_{ptr}^{cov}$						T2T						BE2T						BART					
	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg
CNN	68.6	71.1	73.3	69.9	53.9	67.4	89.4	91.3	92.2	91.7	83.5	89.6	95.9	94.5	90.9	96.9	94.6	94.6	72.4	75.7	71.5	71.8	70.5	72.4	78.7	83.9	87.7	92.1	78.7	84.2	69.9	77.9	87.4	84.1	90.2	81.9
XSUM	13.4	23.5	18.1	13.2	31.0	19.8	6.3	17.8	9.0	8.2	23.2	12.9	7.4	18.1	11.0	7.6	6.5	10.1	9.7	22.6	10.8	9.9	19.1	14.4	14.5	21.1	29.8	8.7	31.3	21.1	35.5	24.7	36.1	50.1	50.7	39.4
Pubm.	61.0	70.0	62.8	78.6	46.6	63.8	77.6	80.7	81.5	75.1	85.9	80.2	70.7	75.6	76.6	67.9	75.4	73.2	58.5	59.3	56.2	72.3	34.9	56.2	55.4	58.7	70.8	71.7	56.4	62.6	69.5	61.5	58.4	61.3	94.1	69.0
Patent B	94.4	94.3	89.0	71.9	91.0	88.1	65.2	60.3	70.9	62.8	71.0	66.0	67.0	63.3	64.6	61.6	77.4	66.8	79.2	81.2	84.4	68.7	73.9	77.5	85.4	88.4	80.3	66.5	82.0	80.6	52.1	53.8	69.0	67.4	76.8	63.8
Red.	20.9	40.2	11.1	13.2	50.9	27.3	37.2	21.5	55.2	62.6	61.1	47.5	27.4	23.5	42.9	49.7	62.2	41.1	34.8	35.7	50.6	44.6	52.5	43.6	17.2	25.7	25.1	30.0	50.3	29.6	59.6	50.3	69.1	49.3	44.2	54.5
avg	51.7	59.8	50.9	49.4	54.7	53.3	55.2	54.3	61.8	60.1	65.0	59.2	53.7	55.0	57.2	56.7	63.2	57.2	50.9	54.9	54.7	53.5	50.2	52.8	50.2	55.6	58.7	53.8	59.8	55.6	57.3	53.6	64.0	62.4	71.2	61.7

Table 8: factcc result for Abstractive models

EXT models	LSTM$_{non}$						Trans$_{non}$						Trans$_{auto}$						BERT$_{non}$						BERT$_{match}$					
	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg	CNN.	XSUM	Pubm.	Patent B	Red.	avg
CNN	99.2	99.9	96.0	99.1	95.2	97.9	100.0	100.0	98.0	99.1	100.0	99.4	98.1	100.0	91.3	93.5	100.0	96.6	99.6	99.9	97.3	98.2	98.6	98.7	99.8	99.4	92.9	95.7	99.1	97.4
XSUM	84.1	94.3	90.3	81.4	94.1	88.9	99.8	100.0	97.4	98.2	100.0	99.1	86.8	99.3	82.9	69.9	100.0	87.8	98.4	99.7	96.6	95.7	99.9	98.1	99.7	99.5	93.2	95.1	98.8	97.3
Pubm.	70.5	84.3	80.8	65.1	89.0	77.9	97.7	98.8	95.1	94.7	100.0	97.3	87.5	99.6	79.0	64.4	99.7	86.1	95.3	99.3	95.1	94.3	99.5	96.7	99.7	99.2	93.1	95.2	99.3	97.3
Patent B	86.1	96.0	90.9	74.1	96.0	88.6	98.3	99.8	96.3	97.4	99.5	98.3	90.7	99.8	85.5	68.8	99.7	88.9	97.0	99.0	96.0	94.8	99.1	97.2	99.7	99.0	93.0	94.5	98.4	96.9
Red.	81.0	92.1	86.9	64.6	90.2	83.0	90.3	94.1	94.1	86.7	96.3	92.3	79.4	98.7	79.6	56.4	98.1	82.5	97.0	98.9	95.3	91.9	98.8	96.4	99.7	99.3	93.1	96.1	99.3	97.5
avg	84.2	93.3	89.0	76.8	92.9	87.2	97.2	98.6	96.2	95.2	99.2	97.3	88.5	99.5	83.7	70.6	99.5	88.4	97.5	99.4	96.1	95.0	99.2	97.4	99.7	99.3	93.0	95.3	99.0	97.3

Table 9: factcc result for Extractive models

analysis aspect			Architecture					
model type			ABS					
compare models			L2L$_{ptr}$ vs. L2L					
holistic analysis			stiff. : 20.74 vs. 18.03 stable. : 68.63 vs. 66.93					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	4.3	0.5	5.3	3.2	1.5	3.0
		Xsum	3.4	1.4	3.4	4.2	0.1	2.5
		Pubm.	10.3	2.3	4.2	3.0	2.6	4.5
		Patent b	1.1	-1.1	2.5	0.6	-0.3	0.5
		Red.	2.2	3.1	2.6	2.9	4.4	3.0
		avg	4.2	1.2	3.6	2.8	1.7	2.7
	normali.	CNN.	0.0	-1.0	4.8	8.7	-9.9	0.5
		Xsum	1.8	0.0	0.7	12.2	-13.8	0.2
		Pubm.	23.3	5.6	0.0	8.4	1.6	7.8
		Patent b	-1.6	-5.8	-0.4	0.0	-14.5	-4.4
		Red.	1.9	8.7	3.4	8.4	0.0	4.5
		avg	5.1	1.5	1.7	7.5	-7.3	1.7

analysis aspect			Architecture					
model type			ABS					
compare models			L2L$_{ptr}^{cov}$ vs. L2L$_{ptr}$					
holistic analysis			stiff. : 22.81 vs. 20.74 stable. : 70.71 vs. 68.63					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	2.9	1.8	6.4	3.4	1.7	3.2
		Xsum	-0.8	-0.8	-4.5	-2.4	-0.1	-1.7
		Pubm.	4.5	1.7	3.2	3.4	2.7	3.1
		Patent b	1.0	2.0	2.2	4.9	0.8	2.2
		Red.	3.3	1.0	6.5	6.9	-0.0	3.5
		avg	2.2	1.1	2.8	3.2	1.0	2.1
	normali.	CNN.	0.0	8.1	9.6	-4.1	8.0	4.3
		Xsum	-6.7	0.0	-19.7	-18.0	-0.2	-8.9
		Pubm.	6.7	7.4	0.0	-1.2	12.6	5.1
		Patent b	-0.1	8.1	0.8	0.0	3.7	2.5
		Red.	5.6	4.9	14.8	11.7	0.0	7.4
		avg	1.1	5.7	1.1	-2.3	4.8	2.1

analysis aspect			Architecture					
model type			ABS					
compare models			T2T vs. L2L					
holistic analysis			stiff. : 19.79 vs. 18.03 stable. : 62.12 vs. 66.93					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	7.1	2.4	7.2	5.2	5.4	5.4
		Xsum	0.4	0.7	-5.4	-4.5	-0.6	-1.9
		Pubm.	1.6	-1.5	2.9	2.9	2.1	1.6
		Patent b	1.2	-2.5	3.0	12.5	-0.0	2.9
		Red.	2.0	0.1	-0.9	-0.7	3.1	0.7
		avg	2.5	-0.1	1.3	3.1	2.0	1.8
	normali.	CNN.	0.0	6.8	15.0	-14.3	14.5	4.4
		Xsum	-10.7	0.0	-24.9	-31.2	-13.4	-16.0
		Pubm.	-2.9	-6.1	0.0	-13.8	2.7	-4.0
		Patent b	-3.8	-9.6	4.2	0.0	-9.9	-3.8
		Red.	-1.0	-0.4	-6.4	-15.2	0.0	-4.6
		avg	-3.7	-1.9	-2.4	-14.9	-1.2	-4.8

analysis aspect			Architecture					
model type			ABS					
compare models			BE2T vs. T2T					
holistic analysis			stiff. : 23.49 vs 19.79 stable. : 62.93 vs. 62.12					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	1.4	0.2	1.1	1.4	1.1	1.0
		Xsum	2.1	10.0	3.2	4.3	5.1	4.9
		Pubm.	6.7	3.1	6.4	8.4	1.4	5.2
		Patent b	1.1	0.4	2.0	0.2	2.4	1.2
		Red.	5.6	3.5	7.9	7.0	6.7	6.1
		avg	3.4	3.4	4.1	4.3	3.3	3.7
	normali.	CNN.	0.0	-17.1	-14.9	2.9	-20.1	-9.8
		Xsum	3.3	0.0	-0.4	9.9	5.5	3.7
		Pubm.	14.9	-1.6	0.0	19.4	-8.5	4.8
		Patent b	1.4	-8.4	-6.0	0.0	-4.6	-3.5
		Red.	12.3	0.2	16.0	16.2	0.0	8.9
		avg	6.4	-5.4	-1.1	9.7	-5.5	0.8

analysis aspect			Architecture					
model type			ABS					
compare models			BART vs. BE2T					
holistic analysis			stiff. : 31.66 vs. 23.49 stable. : 73.83 vs. 62.93					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	3.7	2.0	1.3	1.8	0.8	1.9
		Xsum	4.5	6.0	7.4	9.7	5.8	6.7
		Pubm.	14.1	9.2	7.4	4.1	6.1	8.2
		Patent b	17.6	12.9	12.1	2.7	7.9	10.6
		Red.	17.7	10.0	18.5	14.3	6.6	13.4
		avg	11.5	8.0	9.3	6.5	5.4	8.2
	normali.	CNN.	0.0	-2.5	-11.9	-0.9	-12.9	-5.6
		Xsum	5.2	0.0	8.1	18.2	5.5	7.4
		Pubm.	27.1	15.6	0.0	4.3	9.1	11.2
		Patent b	36.1	24.8	17.2	0.0	13.8	18.4
		Red.	35.5	17.7	33.9	28.4	0.0	23.1
		avg	20.8	11.1	9.4	10.0	3.1	10.9

analysis aspect			Architecture					
model type			ABS					
compare models			BART vs. L2L					
holistic analysis			stiff. : 31.66 vs. 18.03 stable. : 73.83 vs. 66.93					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	12.3	4.5	9.6	8.3	7.3	8.4
		Xsum	7.0	16.7	5.1	9.6	10.3	9.7
		Pubm.	22.4	10.8	16.7	15.4	9.7	15.0
		Patent b	19.9	10.9	17.1	15.4	10.2	14.7
		Red.	25.3	13.7	25.5	20.7	16.4	20.3
		avg	17.4	11.3	14.8	13.9	10.8	13.6
	normali.	CNN.	0.0	-12.8	-11.9	-12.2	-18.4	-11.1
		Xsum	-2.2	0.0	-17.1	-3.1	-2.4	-5.0
		Pubm.	39.2	7.9	0.0	10.0	3.4	12.1
		Patent b	33.7	6.8	15.4	0.0	-0.7	11.0
		Red.	46.8	17.4	43.4	29.3	0.0	27.4
		avg	23.5	3.9	6.0	4.8	-3.6	6.9

analysis aspect			Architecture					
model type			ABS					
compare models			BART vs. T2T					
holistic analysis			stiff. : 31.66 vs. 19.79 stable. : 73.83 vs. 62.12					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	5.2	2.1	2.4	3.1	1.9	3.0
		Xsum	6.6	16.0	10.6	14.1	10.9	11.6
		Pubm.	20.8	12.3	13.8	12.5	7.5	13.4
		Patent b	18.7	13.3	14.0	2.9	10.3	11.8
		Red.	23.3	13.5	26.4	21.3	13.3	19.6
		avg	14.9	11.5	13.4	10.8	8.8	11.9
	normali.	CNN.	0.0	-19.6	-26.9	2.1	-32.9	-15.5
		Xsum	8.4	0.0	7.7	28.1	11.0	11.1
		Pubm.	42.1	14.0	0.0	23.7	0.6	16.1
		Patent b	37.4	16.4	11.2	0.0	9.2	14.9
		Red.	47.8	17.9	49.8	44.6	0.0	32.0
		avg	27.1	5.7	8.4	19.7	-2.4	11.7

analysis aspect			Architecture					
model type			EXT					
compare models			Trans$_{non}$ vs. LSTM$_{non}$					
holistic analysis			stiff. : 28.02 vs. 28.51 stable. : 99.05 vs. 87.00					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	-0.5	-0.8	-1.8	-0.8	13.8	2.0
		Xsum	3.2	-3.8	-2.5	4.2	2.9	0.8
		Pubm.	4.5	-1.6	-4.5	-0.7	-3.0	-1.0
		Patent b	3.9	0.9	-2.6	-4.9	-2.2	-1.0
		Red.	-4.4	-2.0	-3.4	-1.9	-4.2	-3.2
		avg	1.3	-1.5	-2.9	-0.8	1.5	-0.5
	normali.	CNN.	0.0	16.7	5.8	9.2	104.9	27.3
		Xsum	8.9	0.0	4.5	22.8	37.1	14.6
		Pubm.	12.0	11.0	0.0	9.9	4.4	7.4
		Patent b	10.5	25.2	4.2	0.0	7.4	9.5
		Red.	-9.7	8.0	2.4	6.3	0.0	1.4
		avg	4.3	12.2	3.4	9.6	30.7	12.1

analysis aspect			Architecture					
model type			EXT					
compare models			Trans$_{auto}$ vs. Trans$_{non}$					
holistic analysis			stiff. : 28.51 vs. 28.02 stable. : 88.71 vs. 99.05					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	0.5	0.4	3.1	0.9	-12.9	-1.6
		Xsum	-3.2	3.5	4.6	-5.7	-2.3	-0.6
		Pubm.	-0.7	1.2	4.3	2.2	0.1	1.4
		Patent b	-2.9	0.1	3.9	4.4	1.4	1.4
		Red.	1.2	1.7	4.5	-0.5	2.3	1.8
		avg	-1.0	1.4	4.1	0.3	-2.3	0.5
	normali.	CNN.	0.0	-17.7	-2.1	-7.9	-91.9	-23.9
		Xsum	-9.0	0.0	1.0	-25.6	-26.0	-11.9
		Pubm.	-2.8	-12.2	0.0	-4.7	-10.9	-6.1
		Patent b	-8.0	-19.4	-0.6	0.0	-3.6	-6.3
		Red.	2.0	-8.8	0.7	-11.2	0.0	-3.4
		avg	-3.6	-11.6	-0.2	-9.9	-26.5	-10.3

analysis aspect			Architecture					
model type			EXT					
compare models			BERT$_{match}$ vs. BERT$_{non}$					
holistic analysis			stiff. : 32.27 vs. 28.98 stable. : 91.98 vs. 88.93					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	1.6	4.1	4.5	3.0	4.7	3.6
		Xsum	2.9	3.2	3.5	1.6	5.7	3.4
		Pubm.	0.9	4.0	2.4	0.2	8.7	3.3
		Patent b	4.6	3.1	3.5	3.0	3.7	3.6
		Red.	3.3	4.2	3.5	-1.4	3.5	2.6
		avg	2.6	3.7	3.5	1.3	5.3	3.3
	normali.	CNN.	0.0	5.8	5.3	0.7	6.9	3.7
		Xsum	3.4	0.0	2.8	-2.7	11.5	3.0
		Pubm.	-1.2	6.1	0.0	-6.5	26.5	5.0
		Patent b	7.3	1.8	2.8	0.0	3.3	3.0
		Red.	4.4	6.2	2.9	-10.5	0.0	0.6
		avg	2.8	4.0	2.7	-3.8	9.6	3.1

analysis aspect			Architecture					
model type			EXT					
compare models			BERT$_{non}$ vs. Trans$_{non}$					
holistic analysis			stiff. : 28.98 vs. 28.02 stable. : 88.93 vs. 99.05					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	1.8	1.2	0.3	0.8	-10.9	-1.3
		Xsum	-0.9	6.0	0.1	-1.6	-0.7	0.6
		Pubm.	2.5	1.4	0.3	0.6	-2.2	0.5
		Patent b	0.5	1.1	0.2	1.4	3.8	1.4
		Red.	8.3	3.0	-0.1	1.6	5.6	3.7
		avg	2.4	2.5	0.2	0.6	-0.9	1.0
	normali.	CNN.	0.0	-23.9	0.1	-1.5	-96.6	-24.4
		Xsum	-6.1	0.0	-0.5	-8.3	-31.8	-9.3
		Pubm.	2.0	-21.0	0.0	-2.2	-33.7	-11.0
		Patent b	-2.6	-24.8	-0.2	0.0	-5.5	-6.6
		Red.	16.3	-12.8	-1.0	1.0	0.0	0.7
		avg	1.9	-16.5	-0.3	-2.2	-33.5	-10.1

analysis aspect			Generation way					
model type			LSTM					
compare models			LSTM$_{non}$ vs. L2L					
holistic analysis			stiff. : 28.51 vs. 18.03 stable. : 87.00 vs. 66.93					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	8.6	0.1	13.2	4.9	2.0	5.7
		Xsum	13.1	-8.8	18.3	7.1	3.8	6.7
		Pubm.	18.6	4.8	15.1	11.1	9.0	11.7
		Patent b	19.7	2.8	22.8	8.8	5.9	12.0
		Red.	21.4	7.3	30.7	18.0	3.6	16.2
		avg	16.3	1.2	20.0	10.0	4.9	10.5
	normali.	CNN.	0.0	28.4	-0.7	-7.9	-4.8	3.0
		Xsum	18.3	0.0	15.8	2.0	6.6	8.5
		Pubm.	36.8	44.3	0.0	12.5	35.1	25.7
		Patent b	39.6	35.2	31.4	0.0	17.8	24.8
		Red.	44.7	52.9	58.4	35.1	0.0	38.2
		avg	27.9	32.2	21.0	8.3	10.9	20.1

analysis aspect			Generation way					
model type			BERTSUM					
compare models			Trans$_{non}$ vs. T2T					
holistic analysis			stiff. : 28.02 vs. 19.79 stable. : 99.05 vs. 62.12					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	0.9	-3.1	4.3	-1.1	10.3	2.3
		Xsum	15.9	-13.3	21.3	15.8	7.3	9.4
		Pubm.	21.4	4.7	7.7	7.6	3.9	9.1
		Patent b	22.3	6.1	17.2	-8.5	3.7	8.2
		Red.	15.0	5.1	28.2	16.8	-3.7	12.3
		avg	15.1	-0.1	15.7	6.1	4.3	8.2
	normali.	CNN.	0.0	38.3	-9.8	15.6	85.6	25.9
		Xsum	37.8	0.0	45.1	56.1	57.2	39.2
		Pubm.	51.6	61.4	0.0	36.1	36.7	37.2
		Patent b	53.9	70.1	31.5	0.0	35.0	38.1
		Red.	36.1	61.4	67.2	56.6	0.0	44.3
		avg	35.9	46.2	26.8	32.9	42.9	36.9

analysis aspect			Generation way					
model type			Transformer					
compare models			BERT$_{non}$ vs. BE2T					
holistic analysis			stiff. : 28.98 vs. 23.49 stable. : 88.93 vs. 62.93					
fine-grain analysis			CNN.	Xsum	Pubm.	Patent b	Red.	avg
ROUGE	origin	CNN.	1.3	-2.0	3.5	-1.8	-1.7	-0.1
		Xsum	12.9	-17.2	18.3	9.9	1.5	5.1
		Pubm.	17.2	2.9	1.6	-0.3	0.3	4.3
		Patent b	21.8	6.7	15.4	-7.2	5.1	8.4
		Red.	17.8	4.6	20.2	11.4	-4.8	9.8
		avg	14.2	-1.0	11.8	2.4	0.1	5.5
	normali.	CNN.	0.0	31.5	5.2	11.1	9.0	11.4
		Xsum	28.4	0.0	45.0	37.8	19.9	26.2
		Pubm.	38.7	42.0	0.0	14.5	11.5	21.3
		Patent b	49.9	53.7	37.2	0.0	34.1	35.0
		Red.	40.1	48.4	50.2	41.5	0.0	36.1
		avg	31.4	35.1	27.5	21.0	14.9	26.0

Table 10: The difference of ROUGE-1 F1 scores between different models pairs. Every column of the table represents the compared result of one pair of models. The line of holistic analysis displays the overall stiffness and stableness of compared models. The rest of the table is the fine-grained results, the first and third lines of which are the origin compared result ($\mathbf{U_A} - \mathbf{U_B}$ for models pairs A and B) and the second and fourth lines are the normalized compared result ($\hat{\mathbf{U}}_\mathbf{A} - \hat{\mathbf{U}}_\mathbf{B}$ for models pairs A and B). For all heatmap, 'grey' represents positive, 'red' represents negative and 'white' represents approximately zero.