A Closer Look at Data Bias in Neural Extractive Summarization Models

Ming Zhong[*], **Danqing Wang**[*], **Pengfei Liu**[*], **Xipeng Qiu**[†], **Xuanjing Huang**
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, China
{mzhong18,dqwang18,pfliu14,xpqiu,xjhuang}@fudan.edu.cn

Abstract

In this paper, we take stock of the current state of summarization datasets and explore how different factors of datasets influence the generalization behaviour of neural extractive summarization models. Specifically, we first propose several properties of datasets, which matter for the generalization of summarization models. Then we build the connection between priors residing in datasets and model designs, analyzing how different properties of datasets influence the choices of model structure design and training methods. Finally, by taking a typical dataset as an example, we rethink the process of the model design based on the experience of the above analysis. We demonstrate that when we have a deep understanding of the characteristics of datasets, a simple approach can bring significant improvements to the existing state-of-the-art model.

1 Introduction

Neural network-based models have achieved great success on summarization tasks (See et al., 2017; Celikyilmaz et al., 2018; Jadhav and Rajan, 2018). Current studies on summarization either explore the possibility of optimization in terms of **networks' structures** (Zhou et al., 2018; Chen and Bansal, 2018; Gehrmann et al., 2018), the improvement in terms of **training schemas** (Wang et al., 2019; Narayan et al., 2018; Wu and Hu, 2018; Chen and Bansal, 2018), or the information fusion with large **pre-trained knowledge** (Peters et al., 2018; Devlin et al., 2018; Liu, 2019; Dong et al., 2019). More recently, Zhong et al. (2019) conducts a comprehensive analysis on why existing summarization systems perform so well from above three aspects. Despite their success, a relatively missing topic[1] is to analyze and understand the impact on the models' generalization ability from a dataset perspective. With the emergence of more and more summarization datasets (Sandhaus, 2008; Nallapati et al., 2016; Cohan et al., 2018; Grusky et al., 2018), the time is ripe for us to bridge the gap between the insufficient understanding of the nature of *datasets* themselves and the increasing improvement of the *learning methods*.

In this paper, we take a step towards addressing this challenge by taking neural extractive summarization models as an interpretable testbed, investigating how to quantify the characteristics of datasets. As a result, we could explain the behaviour of our models and design new ones. Specifically, we seek to answer two main questions:

Q1: In the summarization task, different datasets present diverse characteristics, so *what is the bias introduced by these dataset choices and how does it influence the model's generalization ability?* We explore two types of factors: *constituent factors* and *style factors*, and analyze how they affect the generalization of neural summarization models respectively. These factors can help us diagnose the weakness of existing models.

Q2: *How different properties of datasets influence the choices of model structure design and training schemas?* We propose some measures and examine their abilities to explain how different model architectures, training schemas, and pre-training strategies react to various properties of datasets.

Our contributions can be summarized as follows:

Main Contributions 1) For the summarization task itself, we diagnose the weakness of existing learning methods in terms of networks' structures, training schemas, and pre-trained knowledge. Some observations could instruct future researchers

[*] These three authors contributed equally.

[†] Corresponding author.

[1] Concurrent with our work, (Jung et al., 2019) makes a similar analysis on datasets biases and presents three factors which matter for the text summarization task.

Proceedings of the 2nd Workshop on New Frontiers in Summarization, pages 80–89
Hong Kong, China, November 4, 2019. ©2019 Association for Computational Linguistics

Factors of Datasets	Measures	Model designs
Constituent[4.1]	Positional coverage rate [4.1.1] Content coverage rate [4.1.2]	Architecture designs [6.2] Pre-trained strategies [6.2]
Style [4.2]	Density [4.2.1] Compression [4.2.2]	Training schemas [6.1]

Table 1: Organization structure of this paper: four measures presented in this paper and choices of model designs they have influence on.

for a new state-of-the-art performance. 2) We show that a comprehensive understanding of the dataset's properties guides us to design a more reasonable model. We hope to encourage future research on how characteristics of datasets influence the behavior of neural networks.

We summarize our observations as follows: 1) Existing models under-utilize the nature of the training data. We demonstrate that a simple training method on CNN/DM (dividing training set based on domain) can achieve significant improvement. 2) BERT is not a panacea and will fail in some situation. The improvement brought by BERT is related to the *style factor* defined in this paper. 3) It is difficult to handle the hard cases (defined by *style factor*) via architecture design and pre-training knowledge under the extractive framework. 4) Based on the sufficient understanding of the nature of datasets, a more reasonable data partitioning (based on *constituent factors*) method can be mined.

2 Related Work

We briefly outline connections and differences to following related lines of research.

Neural Extractive Summarization Recently, neural network-based models have achieved great success in extractive summarization. (Celikyilmaz et al., 2018; Jadhav and Rajan, 2018; Liu, 2019). Existing works on text summarization can roughly fall into one of three classes: exploring *networks' structures* with suitable bias (Cheng and Lapata, 2016; Nallapati et al., 2017; Zhou et al., 2018); introducing new *training schemas* (Narayan et al., 2018; Wu and Hu, 2018; Chen and Bansal, 2018) and incorporating large *pre-trained knowledge* (Peters et al., 2018; Devlin et al., 2018; Liu, 2019; Dong et al., 2019). Instead of exploring the possibility for a new state-of-the-art along one of above three lines, in this paper, we aim to bridge the gap between the lack of understanding of the characteristics for the datasets and the increasing development of above three learning methods.

Concurrent with our work, (Jung et al., 2019) conducts a quite similar analysis on datasets biases and proposes three factors which matter for the text summarization task. One major difference between these two works is that we additionally focus on how dataset biases influence the designs of models.

Understanding the Generalization Ability of Neural Networks While neural networks have shown superior generalization ability, yet it remains largely unexplained. Recently, some researchers begin to take a step towards understanding the generalization behaviour of neural networks from the perspective of network architectures or optimization procedure (Schmidt et al., 2018; Baluja and Fischer, 2017; Zhang et al., 2016; Arpit et al., 2017). Different from these work, in this paper, we claim that interpreting the generalization ability of neural networks is built on a good understanding of the characteristic of the data.

3 Learning Methods and Datasets

3.1 Learning Methods

Generally, given a dataset $\mathcal{D}$, different learning methods are trying to explain the data in diverse ways, which show different generalization behaviours. Existing learning methods for extractive summarization systems vary in architectures designs, pre-trained strategies and training schemas.

Architecture Designs Architecturally speaking, most of existing extractive summarization systems consists of three major modules: **sentence encoder**, **document encoder** and **decoder**.

In this paper, our architectural choices vary with two types of document encoders: `LSTM`[2] (Hochreiter and Schmidhuber, 1997) and `Transformer` (Vaswani et al., 2017) while we keep the sentence encoder (convolutional neural networks) and decoder (sequence labeling) unchanged[3]. The base

[2] We use the implementation of He et al. (2017).

[3] Since they do not show significant influence on our explored experiments.

model in all experiments refers to Transformer equipped with sequence labelling.

Pre-trained Strategies To explore how different pre-trained strategies influence the model, we take two types of pre-trained knowledge into consideration: we choose Word2vec (Mikolov et al., 2013) as an investigated exemplar for non-contextualized word embeddings and adopt BERT as a contextualized word pre-trainer (Devlin et al., 2018).

Training Schemas In general, we train a monolithic model to fit the dataset, but in particular, when the data itself has some special properties, we can introduce different training methods to fully exploit all the information contained in the data.

1. **Multi-domain Learning** The basic idea of multi-domain learning in this paper is to introduce domain tag as a low-dimension vector which can augment learned representations. Domain-aware model will make it possible to learn domain-specific features.

2. **Meta-learning** we also try to make models aware of different distribution by meta-learning. Specifically, for each iteration, we sample several domains as meta-train and the other as meta-test. The meta-test gradients will be combined with the meta-train gradients and finally update the model.

3.2 Datasets

We explore four mainstream news articles summarization datasets (`CNN/DM`, `Newsroom`, `NYT50` and `DUC2002`) which are various in their publications. We also modify two large-scale scientific paper datasets (`arXiv` and `PubMed`) to investigate characteristics for different domains. Detailed statistics are illustrated in Table 2.

4 Quantifying Characteristics of Text Summarization Datasets

In this paper, we present four measures to quantify the characteristics of summarization datasets, which can be abstracted into two types: *constituent factor* and *style factors*.

4.1 Constituent Factors

Motivation When the neural summarization model determines whether a sentence should be extracted, the representation of the sentence consists of two components: position representation[4], which indicates the position of the sentence in the document; content representation, which contains the semantic information of the sentence.

Therefore, we define the position and content information of the sentence as constituent factors, aiming to explore how the selected sentences in the test set relate to the training set in terms of position and content information.

4.1.1 Positional Information

Positional Value (P-Value) Given a document $D = s_1, \cdots, s_n$, for each sentence s_i with label $y_i = 1$, we introduce the notion of positional value $p_i \in 1, \cdots, K$, whose value is the output of the mapping function $p_i = \mathrm{f}(i)$.

Positional Coverage Rate (PCR) Taking positional value p as a discrete random variable, we can define the discrete probability distribution of p over a dataset $\mathcal{D}$,

$$P(p = u) = \frac{N_u}{N_{sent}} \tag{1}$$

where N_u denotes the number of sentence with $p = u$ and N_{sent} represents the number of sentences with $y_i = 1$ in dataset $\mathcal{D}$.

Based on above definition, for any two datasets $\mathcal{D}^A$ and $\mathcal{D}^B$, we could quantify the proximity of their positional value distribution

$$\eta_p(\mathcal{D}^A, \mathcal{D}^B) = -log(\mathrm{KL}(P^A || P^B)) \tag{2}$$

where $\mathrm{KL}(\cdot)$ denotes KL-divergence function. P^A and P^B represent two position value distribution over two datasets. The datasets with similar positional value distribution usually have large PCR η_p.

4.1.2 Content Information

Content Value (C-Value) Given a dataset $\mathcal{D}$, we want to find the patterns that appear most frequently in the ground truth[5] of $\mathcal{D}$ and score them. For each sentence in gound truth, we remove the stop words and punctuation, replace all numbers with "0", and perform lemmatization on each token. After the pre-processing, we treat n-gram ($n > 1$) as the pattern in $\mathcal{D}$ and calculate the score $\varphi(pt_i, \mathcal{D})$ for

[4] The position representation is obtained from the model structure in LSTM and by positional embedding in Transformer.

[5] Ground truth is extracted by the greedy algorithm in Nallapati et al. (2017)

	Statistics			Measures		Lead-k			Ext-Oralce		
	Train	Valid	Test	Density	Compres.	R-1	R-2	R-L	R-1	R-2	R-L
CNN/DM (3)	287,227	13,368	11,490	3.70	13.76	40.24	17.53	36.29	56.55	33.40	53.03
arXiv (6)	187,324	6,218	6,217	2.19	5.59	35.37	9.25	30.93	52.44	22.72	46.15
PubMed (5)	87,897	4,946	5,031	2.04	2.28	36.09	11.49	32.13	46.19	19.91	40.83
DUC2002 (6)	-	-	567	4.43	5.52	47.65	23.19	43.92	62.15	37.30	58.33
NYT50 (4)	96,826	4,000	3,452	4.64	15.33	38.54	19.90	35.27	63.97	43.51	60.70
Newsroom (2)	995,041	108,837	108,862	8.60	35.07	34.19	23.42	31.41	54.97	41.28	51.62

Table 2: Detailed statistics of six datasets. Density and Compression are *style factors* in Section 4.2. Lead-k indicates ROUGE score of the first k sentences in the document and Ext-Oracle indicates ROUGE score of sentences in the ground truth, they represent the lower and upper bound of extractive models respectively. The figure in parentheses after the datasets denotes the number of sentences extracted in Lead-k, which is close to the average number of Ext-Oracle labels.

each pattern as follows:

$$\varphi(pt_i, \mathcal{D}) = \frac{N_{pt_i}}{\sum\limits_{pt_j \in \mathcal{D}} N_{pt_j}} \tag{3}$$

where N_{pt_i} denotes the number of i-th pattern.

Content Coverage Rate (CCR) We introduce the notion of η_c to measure the degree of contents' overlap between training and test set in which the sentences with ground truth labels reside in.

$$\mathrm{Sim(i,j)} = \varphi(\mathrm{pt_i}, \phi_{\mathrm{tr}}) * \varphi(\mathrm{pt_j}, \phi_{\mathrm{te}}) \tag{4}$$

$$\eta_c(\mathcal{D}_{tr}, \mathcal{D}_{te}) = \sum_{pt_i \in \phi_{tr}} \sum_{pt_j \in \phi_{te}} \mathrm{Sim(i,j)} \tag{5}$$

where ϕ denotes the set[6] of patterns which is helpful to pick out ground truth sentences. $\mathrm{Sim}(\cdot)$ measures the similarity of two patterns, $\mathcal{D}_{tr}$ and $\mathcal{D}_{te}$ represent the training set and test set of $\mathcal{D}$ respectively.

4.2 Style Factors

Motivation Different from constituent factors, style factors influence the generalization ability of summarization models by adjusting the learning difficulty of samples' features.

For this type of factor, we did not propose a new measure, but adopt the indicators DENSITY, COMPRESSION proposed by (Grusky et al., 2018)[7] We claim that the contribution here is to focus on the understanding of these metrics and explore the reasons why they affect the performance of summarization models, which is missing from previous work. More importantly, only when we understand how these metrics affect the performance of the models can we use them to explain some of the differences in model generalization.

4.2.1 Density

Density is used to qualitatively measure the degree to which a summary is derivative of a document (Grusky et al., 2018). Specifically, given a document D and its corresponding summary S, Density(D,S) measures the percentage of words in the summary that are from document.

$$\mathrm{Density}(D, S) = \frac{1}{|S|} \sum_{f \in \mathcal{F}(D,S)} |f|^2 \tag{6}$$

where $|\cdot|$ denotes the number of words. $\mathcal{F}(D, S)$ is a set of extractive fragments, which characterize the the longest shared token sequence.

4.2.2 Compression

Compression is used to characterize the word ratio between the document and summary (Grusky et al., 2018).

$$\mathrm{Compression}(d, s) = |D| \,/\, |S| \tag{7}$$

5 Investigating Influence of Proposed Factors on Summarization Models

5.1 Constituent Factors

5.1.1 Exp-I: Breaking Down the Test set

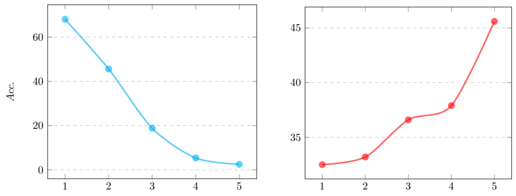
For the P-Value, the threshold set can be denoted as $\{t_0 = 0, t_1, \cdots, t_K = \infty\}$. We calculate $Pos(i)$ for each sentence s_i:

$$Pos(i) = \begin{cases} i & 0 \le i < t_1 \text{ or } i \ge t_{K-1} \\ \frac{i}{n} \cdot t_{K-1} & others \end{cases} \tag{8}$$

and define $p_i = k$ if $t_{k-1} \le Pos(i) < t_k$. The $Pos(i)$ considers both absolute and relevant position of the sentence in the document. In

[6] We choose 100 bigrams and trigrams as the set.

[7] DENSITY and COMPRESSION was originally used to describe the diversity between datasets in the construction of new datasets.



(a) P-Value

(b) C-Value

Figure 1: The accuracy on CNN/DM dataset, test set is broken down based on P-Value and C-Value.

the experiment, we make $K = 5$ and choose $\{0, 3, 7, 15, 35, \infty\}$ for the threshold set.

For the C-Value, we calculate the score for each sentence based on the pattern score from training set.

$$\varphi(\mathrm{s_i}, \mathcal{D}) = \sum_{pt_j \in s_i} \varphi(\mathrm{pt_j}, \mathcal{D}_{\mathrm{tr}}) \tag{9}$$

where s_i denotes sentence in the ground truth of test set. The score indicates the degree of overlap between the sentence and important patterns of the training set. We then sort all the sentences in ascending order by score and divide them into five intervals with the same number of sentences.

As shown in Figure 1, when the sentence is in the front of the document or contains more salient patterns, the accuracy of the model to extract sentences is higher. The phenomenon means that our proposed P-Value and C-Value reflect position distribution and content information of a specific dataset to a certain extent, and the model does learn *constituent factors* and uses them to determine whether a sentence is selected.

5.1.2 Exp-II: Cross-dataset Generalization

From the above experiments, we can see that P-Value and C-Value are sufficient to characterize some attributes in a specific dataset, but beyond that, we seek to understand the differences between mainstream datasets through PCR and CCR.

We calculate PCR/CCR score and measure the performance of the base model by ROUGE-2 score on five datasets. We can see from Table 3 that the training and test set of the same dataset always have the highest PCR/CCR score, which indicates the distribution between them is the closest based on *consitituent factors*. Furthermore, model performance is also in accord with this trend. Consistency presented by the experiment, on the one hand, illustrates that there are significant shifts between different datasets, which results in performance differences of the model in cross-dataset setting, on the other hand, it reflects that position distribution and content information are the key factors of such dataset-shift.

	Dataset	CNNDM	arXiv	Pubmed	NYT50	Newsr.
PCR	CNNDM	1.41	0.56	0.38	0.75	0.70
	arXiv	0.51	2.38	0.68	0.15	0.28
	Pubmed	0.38	0.66	3.79	0.08	0.41
	NYT50	1.27	0.22	0.23	1.46	1.28
	Newsr.	1.02	0.30	0.40	0.79	4.57
CCR	CNNDM	3.69	0.07	0.89	1.32	1.56
	arXiv	0.05	10.04	0.47	0.03	0.16
	Pubmed	0.72	0.62	11.03	0.51	2.03
	NYT50	1.34	0.07	0.75	3.13	2.12
	Newsr.	1.27	0.21	2.09	1.41	4.21
R-2	CNNDM	18.71	9.55	11.60	21.72	15.89
	arXiv	11.46	16.91	16.21	15.10	15.93
	PubMed	9.68	15.56	16.46	10.39	12.16
	NYT50	17.01	9.62	11.98	25.39	20.52
	Newsr.	17.38	9.42	12.23	20.21	24.59

Table 3: Results of cross-dataset PCR(η^p), CCR(η^c) and ROUGE-2 score. Each cell η^p_{ij} and η^c_{ij} denotes the coverage rate between training dataset (rows) and test dataset (columns). Each cell R-2_{ij} denotes model performance in cross-dataset setting.

After verifying the validity of PCR and CCR, we utilize them to estimate the distance between the real distribution of datasets. For instance, news articles datasets (CNN/DM, NTY50 and Newsroom) and scientific paper datasets (arXiv and PubMed) both have lower scores in terms of two metrics, that is to say, there is a larger shift between them, which is also in line with our knowledge. Based on the estimation, we can understand more deeply the impact of different datasets on the generalization ability of various neural extractive summarization models.

5.2 Style Factors

We integrate training set, validation set and test set as a whole set and divide it into three parts according to the density or compression of each article and name them “low”, “medium” and “high”. For example, articles in “density, high” represents these articles have a higher density in the entire dataset. Based on above operation, we break down the test set and attempt to analyze how style factors influence the model performance.

Exploration of Density Density represents the overlap between the summary and the original text, so the samples with high density are more friendly to extractive models. Consequently, it is easy for us to understand the higher the density, the higher the

	Metrics	Low	Medium	High
DENSITY	R-1	35.07	41.63	**46.48**
	R-2	11.22	17.87	**26.39**
	R-L	31.19	37.69	**43.21**
	F1	33.81	38.02	**38.38**
COMPRES.	R-1	**44.95**	41.32	36.41
	R-2	**21.74**	18.61	15.10
	R-L	**41.09**	37.61	32.98
	F1	**39.82**	37.01	32.23

Table 4: The performance of our base model on CNN/DM dataset, test set is broken down based on DENSITY and COMPRESSION

ROUGE score in Table 4. However, the F_1 value of prediction is also positively correlated with the density, which means that density is closely related to the learning difficulty.

		1	2	3	Total
Low	ψ	0.18	0.15	0.13	0.46
	Pct	8.5%	7.0%	6.0%	21.5%
Medium	ψ	0.24	0.20	0.17	0.61
	Pct	9.0%	7.6%	6.4%	23.0%
High	ψ	0.32	0.27	0.22	0.81
	Pct	10.8%	9.1%	7.5%	27.4%

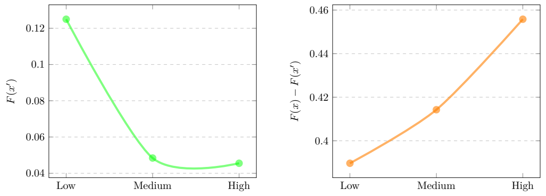
Table 5: Experiment about DENSITY, Pct denotes the percentage of $\psi(s_i, S)$ to $\sum_{s_i \in D} \psi(s_i, S)$. The first three sentences contain more salient information in samples with higher density.

In order to comprehend this correlation, we conduct the following experiment. Given an article and summary pair, we assign a score $\psi(s_i, S)$ to each sentence in article to indicate how much sailent information is contained in the sentence.

$$\psi(s_i, S) = \mathrm{LCS(s_i, S)} \,/\, |s_i| \qquad (10)$$

where $\mathrm{LCS(s_i, S)}$ denotes the longest common subsequence length (not counting stop words and punctuation) of the sentence and summary. We calculate the percentage of $\psi(s_i, S)$ to $\sum_{s_i \in D} \psi(s_i, S)$ and present the results of the three highest-scoring sentences in Table 5. Obviously, in samples with high density, the salient information is more concentrated in a few sentences, making it easier for the model to extract correct sentences.

Therefore, for dataset with high density, we can try to introduce external knowledge into the model, which helps the model better understand the semantic information, and thus easier to capture sentences with salient patterns. In addition, models with external knowledge should have better generalization ablity when transferred to high-density dataset. These inferences will be verified in Section 6.1 and 6.2.1.



(a) Position attribution (b) Content attribution

Figure 2: The position and content attribution on CNN/DM, the test set is broken down based on COMPRESSION.

Exploration of Compression Documents with high compression tend to have fewer sentences because summaries usually have a similar length in the same dataset. So the results of compression in Table 4 are in line with our expectations, how the model represents long documents to get good performance in text summarization task remains a challenge (Celikyilmaz et al., 2018).

Unlike the exploration of density, we attempt to understand how the model extracts sentences when faced with different compression samples. We utilize an attribution technique called Integrated Gradients (IG) (Sundararajan et al., 2017) to separate the position and content information of each sentence. The setting of *input x* and *baseline x'* in this paper is close to Mudrakarta et al. (2018)[8], but it is worth noting that our base model adds positional embedding to each sentence, so *input x* and *baseline x'* both have positional information.

We tend to think that $\mathrm{F(x')}$ denotes the attribution of positional information, and $\mathrm{F(x)}$ - $\mathrm{F(x')}$ denotes the attribution of content information when model makes decisions, where $\mathrm{F} : \mathbb{R}^n \to [0, 1]$ represents a deep network. Figure 2 illustrates that as compression increases, the help provided by positional information is gradually reduced and content information becomes more important to the model. In other words, the model can perceive the compression and decide whether to pay more attention to positional information or important patterns, this observation is helpful for us to design models or study their generalization ability in Section 6.2.1.

[8] Using empty documents (a sequence of word embeddings corresponding to padding value) as *baseline x'*.

Models	DENSITY			COMPRESSION			All
	Low	Medium	High	Low	Medium	High	
LSTM	11.17	17.75	25.84	21.66	18.25	14.73	18.58
- Word2Vec	0.08 ↓	0.36 ↓	0.57 ↓	0.51 ↓	0.32 ↓	0.18 ↓	0.35 ↓
+ BERT	0.44 ↑	0.80 ↑	1.46 ↑	1.19 ↑	0.80 ↑	0.68 ↑	0.92 ↑
Transformer	11.15	17.84	26.17	22.00	18.34	14.65	18.71
- Word2Vec	0.10 ↓	0.30 ↓	0.54 ↓	0.48 ↓	0.12 ↓	0.30 ↓	0.31 ↓
+ BERT	0.52 ↑	0.77 ↑	1.28 ↑	0.98 ↑	0.86 ↑	0.79 ↑	0.88 ↑

Table 6: Performance of models equipped with different types of knowledge on CNN/DM dataset. BERT here removes the gradient as a way of introducing external knowledge.

6 Bridge the Gap between Dataset Bias and Model Design Prior

In this section we investigate how different properties of datasets influence the choices of model structures, pre-trained strategies, and training schemas.

Idea of Experiment Design Through the above analysis in Section 4, the *constituent factors* reflect the relationship between diverse data distributions and *style factors* directly affect the learning difficulty of samples' features. Based on the different attributes of the above two types of factors, we designed the following investigation accordingly: for the *style factor*, we not only investigate the influence of different model architectures and pre-trained strategies on it, but utilize it to explain the generalization behaviour of the models. For the *constituent factors*, we discuss their effects on different training strategies, such as multi-domain learning and meta-learning, because these learning modes are all about how to better model various types of distributions.

6.1 *Style Factors* Bias

In this section, we study **whether the samples with different learning difficulties described by the *style factors* can be well handled through the improvement of structure or the introducing of pre-training knowledge or we need to extend our model in other ways**.

Table 6 shows the breakdown performance on CNN/DM based on DENSITY and COMPRESSION. And we can observe that: 1) An obvious trend is that LSTM performs better than Transformer with increasing difficulty in sample learning (low density and high compression). For instance, LSTM performs worse than Transformer on the subset with high density, while surpasses Transformer when the density of testing examples becomes lower. 2) Generally, the introducing of pre-training word vectors can improve the overall results of the models. However, we found that increasing the learning difficulty of samples would weaken the benefits brought by pre-trained embeddings. 3) The prospects for further gains for these hard cases described by *style factor* from novel architecture design and knowledge pre-training seem quite limited, suggesting that perhaps we should explore other ways, such as generating summaries instead of extracting.

6.2 *Constituent Factors* Bias

We design our experiment towards the answer to two main questions as follows.

6.2.1 Exp-I: How do dataset properties influence the choices of training schemas?

When our training set itself contains multiple domains grouped by the *constituent factors*, how can we make full use of the datasets characteristic and find the most suitable training schemas? For example, CNN/DailyMail, as one of the most popular datasets, consists of two sub-datasets. For this question, dataset-shift discussed in Section 5.1.2 and the learning diffuculties of the dataset should be taken into consideration.

Choices of Training Schemas: We compare four training schemas: joint training, multi-domain learning[9] with explicit information (tag embedding), implicit information (BERT) and meta-learning.

Evaluation Setting: In order to more comprehensively reflect the generalization ability of different models, we conducted zero-shot transfer evaluation. Specifically, each of our models is trained on `CNN/DM` while evaluated both on `CNN/DM` (IN-DATASET) and other datasets (CROSS-DATASET).

[9] We view CNN and DailyMail in CNN/DM as two different domains.

Dataset	Basic			Tag			Meta			Bert		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
CNNDM	41.31	18.71	37.62	41.37	18.89	37.70	41.26	18.77	37.60	**42.27**	**19.72**	**38.62**
Arxiv	36.29	9.55	32.01	**37.12**	10.09	**32.73**	36.25	9.58	31.95	37.00	**10.22**	32.59
Pubmed	36.13	11.60	31.91	36.75	12.02	32.47	36.07	11.65	31.89	**36.93**	**12.15**	**32.71**
DUC2002	49.21	24.08	45.37	49.25	24.28	45.38	49.34	24.22	45.50	**49.77**	**24.67**	**45.87**
NYT50	41.17	21.72	37.85	40.84	21.06	37.45	41.23	21.56	37.89	**43.80**	**24.09**	**40.46**
Newsroom	28.08	15.89	25.08	26.37	13.78	23.21	28.10	15.85	25.06	**29.30**	**16.73**	**26.16**

Table 7: Results of four models under two types of evaluation settings: IN-DATASET, and CROSS-DATASET. Bold indicates the best performance of all models, red indicated the best performance other than BERT.

Table 7 shows the results of four models under two types of evaluation settings: IN-DATASET, and CROSS-DATA, and we have the following findings:

1) For IN-DATASET setting, comparing the *Tag* and the basic models, we find a very simple method that assign each sample a domain tag could achieve improvement. The reason here we claim is that domain-aware model makes full use of the nature of dataset. 2) For multi-domain and meta-learning model, we attempt to explain from the perspective of data distribution. Although meta-learning obtains worse performance under IN-DATASET setting, it yet has achieved impressive performance under CROSS-DATASET setting. Concretely, meta-learning model surpasses *Tag* model on three datasets: `DUC2002`, `NYT50` and `Newsroom`, whose distribution is closer to CNN/DM based on *constituent factors* in Table 3. Correspondingly, *Tag* model uses a randomly initialized embedding for zero-shot transfer, and we suspect that this perturbation unexpectedly generalizes well on some far-distributed datasets (`arXiv` and `PubMed`). 3) BERT has shown its superior performance and nearly outperforms all competitors. However, the generalization ability of BERT is poor on `arXiv`, `PubMed` and `DUC2002` compared to the performance improvement in IN-DATASET setting. In contrast, BERT shows good generalization when tranferring to datasets with high density and compression (`NYT50` and `Newsroom`). As we have discussed in Sec. 5.2, samples with high *style factors* require model to capture salient patterns, which is exactly the improvement of introducing external knowledge from BERT.

Models	R-1	R-2	R-L
Transformer	41.31	18.71	37.62
+ random tag	41.19	18.52	37.57
+ domain tag	**41.41**	18.71	**37.74**
+ P-Value tag	41.38	18.71	37.67
+ C-Value tag	41.39	18.73	37.71
+ P-Value & C-Value tag	**41.41**	**18.74**	**37.74**
Liu (2019)	42.57	19.96	39.04
BERT (our implementation)	42.59	19.92	38.94
+ domain tag	42.72	19.91	39.05
+ P-Value & C-Value tag	**42.77**	**19.98**	**39.10**

Table 8: Results of experiments with tags on our base model and current state-of-the-art model. The usage of BERT here is as same as Liu (2019), which is to fine tune BERT on CNN/DM.

6.2.2 Exp-II: Searching for a Good Domain

The second question we study is **what makes a good domain**? To answer this question, we define the concept of domain based not solely on the dataset, but divide the training set by directly utilizing the *constituent factors*. Specifically, we explore the following different settings:

1) **Random tag:** Each sample is assigned a random "`pseudo-domains`" tag.

2) **Domain:** Divide training samples according to the domain (CNN or DM) they belong to .

3) **P- and C-Value:** Each sentence is assigned a tag by its corresponding P-Value and C-value scores.

We experiment with tags on our base model and the current state-of-the-art model Liu (2019). Liu (2019) and the results are presented in Table 8, we can obtain the following observations:

1) *Random partitioning does not make sense and cannot lead to the improvement of performance.* Conversely, the partitions based on the *constituent factors* have obtained the benefit. 2) This simple learning method that dividing the training set based on domain has shown considerable benefit, which can be complementary to the improvement brought by BERT. 3) The division based on the *constituent factors* (P-value & C-value) achieves the best result in the context of BERT, which implies that *for the summarization task, mining the characteristics of*

the dataset itself plays an important role.

7 Conclusion

In this paper, we conduct a data-dependent understanding of neural extractive summarization models, exploring how different factors of datasets influence these models and how to make full use of the nature of dataset so as to design a more powerful model. Experiments with in-depth analyses diagnose the weakness of existing models and provide guidelines for future research.

References

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 233–242. JMLR. org.

Shumeet Baluja and Ian Fischer. 2017. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387.*

Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. 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 1662–1675.

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)*, 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.*

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197.*

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.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and whats next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 473–483.

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

Aishwarya Jadhav and Vaibhav Rajan. 2018. Extractive summarization with swap-net: Sentences and words from alternating pointer networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 142–151.

Taehee Jung, Dongyeop Kang, Lucas Mentch, and Eduard Hovy. 2019. Earlier isn't always better: Subaspect analysis on corpus and system biases in summarization. *arXiv preprint arXiv:1908.11723.*

Yang Liu. 2019. Fine-tune BERT for extractive summarization. *CoRR*, abs/1903.10318.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the model understand the question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906.

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. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1747–1759.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of NAACL*, volume 1, pages 2227–2237.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. 2018. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pages 5014–5026.

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.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org.

Ashish Vaswani, Noam Shazeer, Jakob Parmar, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in NIPS*.

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*.

Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. 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, Florence, Italy. Association for Computational Linguistics.

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.