TLDR9+; A Large Scale Resource for Extreme Summarization of Social Media Posts

Anonymous EMNLP submission

Abstract

Recent models in developing summarization systems consist of millions of parameters and the model performance is highly dependent on the abundance of training data. While most existing summarization corpora contain data in the order of thousands to one million, generation of large-scale summarization datasets in order of couple of millions is yet to be explored. Practically, more data is better at generalizing the training patterns to unseen data. In this paper, we introduce TLDR9+ — a large-scale summarization dataset— containing over 9 million training instances extracted from Reddit discussion forum ([HTTP]). This dataset is specifically gathered to perform *extreme* summarization (i.e., generating one-sentence summary in high compression and abstraction) and is more than twice larger than the previously proposed dataset. We go one step further and with the help of human annotations, we distill a more fine-grained dataset by sampling **H**igh-**Q**uality instances from TLDR9+ and call it TLDRHQ dataset. We further pinpoint different state-of-the-art summarization models on our proposed datasets.

1 Introduction

Text summarization is defined as generating a concise sequence of text as summary, given relatively a longer document as source. A high-quality summary conveys the most important points of its associated source. The task is generally performed in two ways: 1) extractive, (Nallapati et al., 2017; Dong et al., 2018; Narayan et al., 2020; Cho et al., 2020) in which salient sentences are identified and concatenated to form the final summary; and 2) abstractive, (See et al., 2017; Gehrmann et al., 2018; MacAvaney et al., 2019; Zhang et al., 2019; Lewis et al., 2020; Lebanoff et al., 2020) which is considered more challenging as the model needs to deal with text paraphrasing and novel words generation beyond sentence extraction.

> We go to school together, we have three lessons a week together. She normally sits at the front and I sit at the back, but recently the person I sit next to has been struggling with mental health and hasn't been in, so I moved and sit next to her most lessons. We also do this engineering scheme together, so we have maybe half an hour a week with two other people working on that. For a while now we 've texted each other a few times a week with pictures of our cats, since we both love them. Outside of that, we don't really hang out at all. I see a lot of theatre, and about a week ago she said she wanted to come see something with me. So I agree, I love showing people theatre. When we find our seats, mine has a pole in the way so I can't see a section of the stage unless I lean away from her, but her view is perfect. About half an hour in, she leans on my shoulder. Halfway through act 2 she starts hugging my arm, while still leaning on my shoulder. She was kind of cuddling all day, we went to an arcade earlier as well. She doesn't seem like the cuddling type of friend, and I'm very worried she has a crush on me. I don't want to ruin a friendship, I don't like her back. Should I just ignore it until she asks me? What if she thinks that was a date?
>
> **TL;DR** I took my friend to see a show, she leant on my shoulder the whole time. I 'm not into her but I think she has a crush on me?

Figure 1: An example Reddit post with TLDR summary. As seen, the TLDR summary is extremely short, and highly abstractive.

Over the past few years, different neural models including RNN (Hochreiter and Schmidhuber, 1997) and Transformer-based (Vaswani et al., 2017) networks have been proposed to facilitate the summarization task. While promising, the performance of such models is bound to the abundance of training data due to the massive model complexity (Ying, 2019). Lack of sufficient training data worsens the model's ability to generalize patterns in training data to unseen data (Althnian et al., 2021). In addition, overfitting will be likely inevitable as the model is forced to learn from a limited set of data; hence, hindering the generalization. This justifies the necessity of large-scale corpora for training large and complex models.

Prevalence of social media platforms has provided communities with an opportunity to exchange different types of data while interacting with each other. *Reddit* [1] is one of such popular platforms where users post their content of interest

[1] https://www.reddit.com/

Dataset	Domain	# instances
Non-social media		
SciTLDR	Scientific	3.2K
XSUM	News	227K
Social media		
Reddit TIFU	Social Media	120K
Webis-TLDR-17	Social Media	4M
TLDRHQ (ours)	Social Media	**1.7M**
TLDR9+ (ours)	Social Media	**9.2M**

Table 1: Overview of *extreme* summarization datasets across different social and non-social domains with number of instances.

in a variety of domains. TLDR, acronym for "Too Long; Didn't Read", is a common practice that aims at removing unnecessary information from the lengthy post, and presenting its gist information in a few words. Figure 1 shows a sample of Reddit post with its TLDR, which aims at abstracting post with extreme compression. Abundance of posts that contain such TLDRs during the recent years has given rise to generation of data collections that can be utilized for training deep neural networks; hence, addressing the challenge of large-scale datasets' scarcity. Despite the possibility of acquiring large-scale datasets from social media platforms, training deep neural networks on such datasets is yet challenging. This might be due to the specific writing style of social media content such as *informal* language and massive *noise* within such content (Sotudeh et al., 2020).

Table 1 shows some of the existing summarization datasets in social and non-social media domains. These datasets are specifically proposed for *extreme summarization* task, where the aim is to produce one to two summary sentences in extreme compression and high abstraction. In this paper, we introduce our dataset, TLDR9+ with over 9 millions instances which is more than twice larger than the previous dataset (Völske et al., 2017). We further sample high-quality instances in virtue of human annotations from TLDR9+ to construct TLDRHQ yielding 1.7 million instances in the hope of providing firm grounds for future work. Owing to extremely short length of TLDR summaries (less that 40 words), our datasets are rather suitable for *extreme summarization* task, than for longer ones.

In this research, we aim at harvesting instances that include TLDRs written by the Reddit users spanning the period of 2005-2021. Our early attempt at gathering such instances yields over 9 millions instances with TLDRs as the initial set (i.e., TLDR9+). Since social media posts are inherently noisy, we consider applying a heuristic method to cut out low-quality instances from the initial set, which ultimately results in 1.7 million high-quality instance. For deciding such heuristic, we employ human annotators to help obtaining a more fine-grained dataset (i.e., TLDRHQ). Furthermore, we establish various state-of-the-art extractive and abstractive summarization models on our proposed datasets. Finally, we carry out an analysis over the results on both datasets to shed lights on future direction. We believe that our datasets can be utilized to pave the path for future research. Our miner code and data are made publicly available at [HTTP].

2 Related work

Over the past few years, summarization community has witnessed variety of summarization datasets in different domains (See et al., 2017; Cohan et al., 2018; Kornilova and Eidelman, 2019; Grusky et al., 2018; Sotudeh et al., 2021). While these collections have provided a fair basis to perform different neural text summarization models, the necessity of introducing large-scale collections, in magnitude of over 4 millions, has not been much explored.

Among the first attempts on this track, Rush et al. (2015) gathered the English Gigaword corpus (Graff et al., 2003) which contains around 4 millions article-headline pairs for the task of news headline generation. Researchers have noted that *lead bias* is the common phenomenon in most news datasets, where early parts of the article generally include the most important information (Kedzie et al., 2018; Zhu et al., 2019; Grenander et al., 2019). To alleviate the lead bias for training summarization models, there have been recent efforts to propose summarization datasets, where the lead bias phenomenon is mitigated and summaries are sampled from diverse source regions. Amongst those, Sharma et al. (2019) proposed BIGPATENT, consisting 1.3 million patent documents, collected from Google Patents Public Datasets, with human-written abstractive summaries. Kim et al. (2019) proposed *Reddit TIFU* in which the abstractive gold summaries are sampled from diverse regions of the source document, rather than lead regions.

Our proposed datasets are more suited for the

(a) submission-TLDR (m) comment-TLDR

Figure 2: The proportion of TLDRs over entire posts (submissions and comments) submitted per year (Figures (c) and (d)). At the time of writing this paper, submissions dumps are partly uploaded for 2021 (until 2021-06), while there is no comments dumps uploaded for 2021.

task of *extreme summarization* (Narayan et al., 2018; Cachola et al., 2020), where the task is to create a short one-sentence summary. To this end, Narayan et al. (2018) proposed *XSUM* dataset which is a real-word dataset compiling online articles from the British Broadcasting Corportation (BBC). TLDR generation task is also a new form of extreme summarization. Kim et al. (2019) collected *Reddit-TIFU* dataset, consisting of 120K posts from the online discussions from Reddit. Recent efforts have mined around 4 millions Reddit posts along with their TLDR summaries (Völske et al., 2017) which resulted in Webis-TLDR-17 dataset. While our work is similar to theirs, our collected dataset is more than twice larger than the one previously proposed.

3 The *Reddit* Collection

3.1 Data Collection

Reddit is a social news aggregation, and discussion website platform that has been officially launched since June 2005. It supports some features specific to social platforms such as web content rating though up-voting, and discussion topics via subreddits. The user-created content can be of any domain such as News, Politics, Science, Sport and etc. Users can post or comment on a specific topic which falls into a specific subreddit. Within subreddits, users submit their post as *submission*, and others can react through commenting under the posted submission. Each submission and comment has a *text body/selftext* which reflects the users' information exchange regarding a specific topic. The existence of social platforms such as Reddit has provided the research community with an opportunity to experiment with resources that use *informal* language, rather than those in news, scientific or legal documents which use formal language.

TLDR—Too Long; Didn't Read— is a common practice in Reddit that often appears at the end of long reddit posts. It is denoted as an extremely short summary that urges users to read shorter version of a longer text when they could not be bothered to read the entire posts. Figure 2 shows the ratio of posts containing such TLDR summaries over the entire submitted posts (and comments) across different years. It is observable that although we see an ascending trend since 2005, the number of TLDRs remains fixed (see Section 3.4) while the number of posts increases drastically.

Pushshift [2] is a social media data repository platform that has been recently made available to NLP researchers (Baumgartner et al., 2020). It contains recent and historical dumps of Reddit posts that are updated in real-time. In order to create the TLDR dataset, we downloaded the whole data dumps (submissions and comments) which covers the period of 2005-2021, and extracted instances that contain TLDRs within the posted source text. This mining process resulted in TLDR9+ dataset, that contains over 9 millions instances. To acquire a more fine-grained dataset, with the help of human annotations, we obtained TLDRHQ dataset, consisting of 1.7 millions high-quality instances. The datasets' construction details are discussed in what follows.

[2] https://files.pushshift.io/

3.2 Datasets Construction: TLDR9+ and TLDRHQ

TLDR9+. After downloading Reddit data dumps, we extract posts in which a mention of TLDR-style keywords is found. To find TLDR-style keywords within a given text, we declare a regular expression that matches words starting with "TL" and ending with "DR", with permission of having up to three characters in-between as also done by Völske et al. (2017). This stage yields the **TLDR9+** dataset as the *full* corpus. At the next filtering stage, we utilize a heuristic method along with human supervision to narrow down to a more fine-grained dataset that contain high-quality instances.

TLDRHQ. A few studies have noted that user-generated content in social media platforms is noisy (Liu and Inkpen, 2015). To filter out such noisy instances from the TLDR9+ dataset, we use a heuristic method to drop low-quality instances while retaining high-quality ones. To be more specific, given a post-TLDR pair, we firstly identify the highest score sentence in terms of ROUGE-2 and ROUGE-L mean scores (i.e., *oracle* sentence). We then decide to either drop or retain the instance if the score surpasses a pre-defined *threshold*. We experiment with different thresholds of 0.15, 0.17, 0.20, 0.22 and 0.25, and choose one considering the annotations done by human annotators. The details of human annotation process is discussed in what follows.

3.3 Human Annotation

As mentioned earlier, we first define 5 fixed thresholds including 0.15, 0.17, 0.20, 0.22, and 0.25 to create 5 data subsets from TLDR9+ dataset. Specifically, we take TLDR9+ as the initial seed, from which 5 subsets is created as follows. To gather instances for each of the pre-defined thresholds, we check if the oracle sentence's score in the given instance surpasses the experimented threshold. If it does so, we add it to the subset, otherwise it is dropped. We then randomly sample 20 cases from each of these subsets with their oracle sentence and TLDR summaries, yielding 100 cases for annotation in total. We have four human annotators either confirm (1) or reject (0) if the oracle sentence *validates* the TLDR summary. By definition, the sentence *validates* the TLDR summary if at least one fragment can be found within the sentence that semantically occurs in TLDR summary.

We further provide the instances' text (i.e., source) as the "Context" for the oracle sentence, and ask the annotators to confirm or reject if the context also validates the TLDR summary. Context is specifically important for the cases where the oracle sentence does not validate the TLDR summary. In fact, by providing context, we aspire to verify if an ideal summarizer is able to generate the TLDR using the context when the oracle sentence is not much informative. For tie cases [3], we employ a fifth annotator to make the final decision.

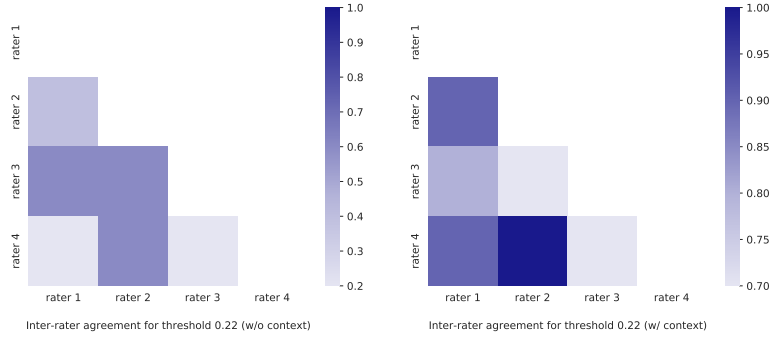
Threshold	score w/o context	score w/ context
0.15	0.65	0.90
0.17	0.90	**1.0**
0.20	0.85	0.95
0.22	**1.0**	**1.0**
0.25	0.75	0.90

Table 2: Average decision scores given by the annotators for each threshold.

Table 2 presents the average *decision score* assigned to the samples on each threshold. The decision score for a given sample is defined as the annotators' average confidence at giving label 1 to that specific sample. If the average confidence score surpasses 0.50, we assign 1 and if it is below 0.50, the sample is annotated with 0. Otherwise, the fifth annotator decides the label. As shown, threshold 0.22 attains the full score in the presence and absence of the context. Overall, this shows that most of the annotators believe the TLDR can be distilled considering both oracle sentence and the entire source.

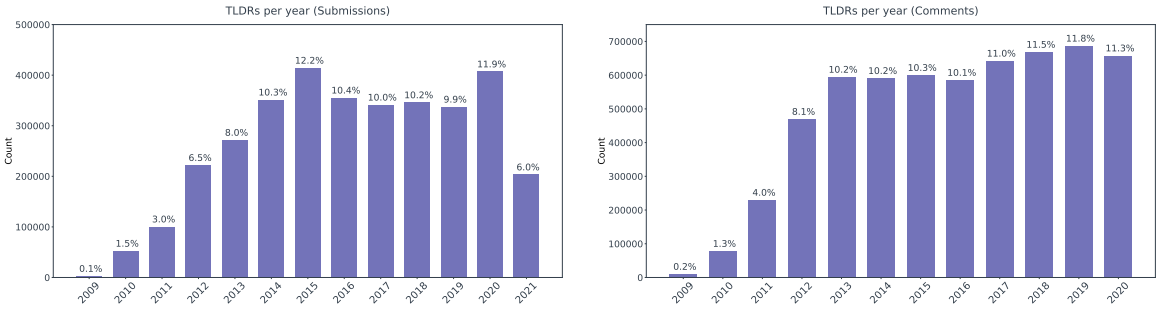
Figure 3 shows pair-wise inter-rater S score agreement (Bennet et al., 1954) throughout the annotation process on threshold 0.22, denoting that annotators have mostly slight or fair agreement in labeling process. Specifically, when the context is not provided (i.e., merely with consideration of oracle sentence), raters (2, 4), (2, 3), and (1, 3) have quite a high rate of agreement. On the other hand, most pairs of annotators including (1, 2), (1, 4), and (2, 4) achieve a high agreement rate when the context is given. As the given decision scores —either only with oracle sentence or provided context— sum up to 1.0, and considering moderately high agreement rate between the annotators, we decide to sample our TLDRHQ dataset from the instances in that was in threshold 0.22's subset. This leads

[3] Suppose a case where two annotators confirm (label 1), while the other two reject (label 0).



(a) Annotation with the oracle sentence (b) Annotation with context

Figure 3: S score inter-rater agreement for annotation without context (left), and annotation with context (right)



(a) submission-TLDR (b) comment-TLDR

Figure 4: The proportion of instances containing TLDR in TLDR9+ dataset. As seen, the number of TLDRs is increasing each year. At the time of conducting this research, the submission data dumps are partially uploaded for 2021 (until 2021-06), while there is no comments uploaded for 2021 in the Pushshift repository.

Dataset	# instances	post (words/sent.)	TLDR (words/sent.)	Compression ratio
TLDR9+	9,227,437	310.3/14.0	35.6/2.3	8.72
TLDRHQ	1,671,099	332.03/15.67	26.96/1.78	12.32

Table 3: Average words and sentence length per instance along with the compression ratio in our proposed datasets.

us to choose human-decided threshold 0.22 as our ground to sample **H**igh-**Q**uality TLDRs for constructing TLDRHQ dataset.

3.4 Dataset Analysis

In this section, we give statistics, along with analyses on the proposed datasets.

Table 3 shows general statistics of datasets in terms of post and TLDR length. As shown, the compression rate [4] is 8.72 and 12.46 in TLDR9+, and TLDRHQ datasets, respectively. This shows that authors generally tend to write much shorter TLDRs that highly shortens the post's text, which is expected due to the nature of TLDR summaries.

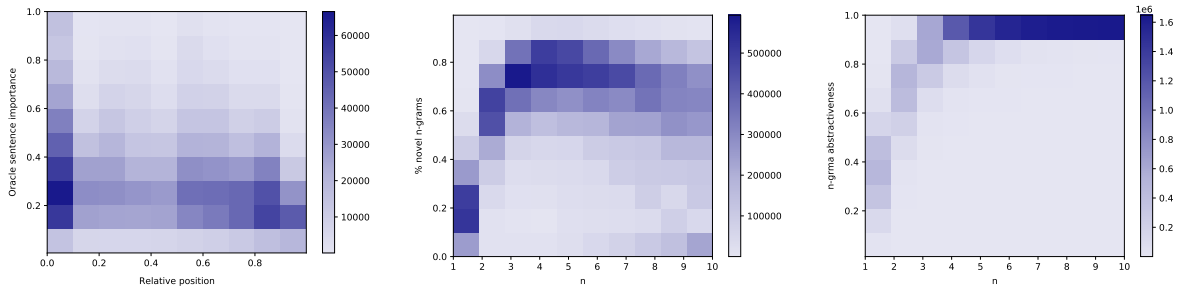
[4] Compression rate = $\frac{\text{Avg post length}}{\text{Avg TLDR length}}$

Figure 4 demonstrates the number of TLDR pairs in TLDR9+ across different years. As observed, 83.65% of these TLDRs occur after 2013 which shows the popularity of this writing style among the Reddit users. We also see a similar trend for years after 2013, each of which constitute a fixed amount (10%-12%) of the dataset.

The oracle sentence's relative position in post's text along with its importance is shown in Figure 5 (a). We define the oracle importance score as follows:

$$\text{oracle importance} = \frac{\max \text{RG}_{2+\text{L}}(s_i)}{\sum_{s_i \in D} \text{RG}_{2+\text{L}}}$$

where D is the set of all sentences within the post, and s_i denotes the ith sentence. $\text{RG}_{2+\text{L}}(.)$ is a function that takes in a post's sentence, and outputs the mean of its ROUGE-2 and ROUGE-L score with respect to TLDR summary. Intuitively, the oracle importance score can be framed as the attention score over the oracle sentences when the scoring function is ROUGE. Observing Figure 5, while more of the oracle sentences occur in early parts of



(a) (b) (c)

Figure 5: Heatmaps of TLDRHQ showing (a) the oracle sentence's importance to its relative position; (b) percentage of novel n-grams; and (c) n-gram abstractiveness. The heat extent shows the number of the instances within the specific bin.

the post's text (< 0.10) with importance score of less than 0.30, it appears that the oracle sentences are spread out across the post's text overall. This observation is substantial, justifying the usability of this dataset for extractive summarization task.

To analyze the abstraction level of TLDRHQ dataset, we plot the percentage of novel n-grams within the TLDR summary (See et al., 2017) in Figure 5 (b), as well as the TLDR's n-gram abstractiveness (Gehrmann et al., 2019) in Figure 5 (c) over the all instances in TLDRHQ dataset. As indicated, there are quite a large proportion of novel n-gram words appeared in the TLDR summary as the heat extent is mostly concentrated in the upper half of the y-axis. These plots show the promising capability and challenges of this dataset to be used for abstractive summarization models.

4 Experimental Setup

4.1 Baselines

We benchmark several extractive and abstractive summarization baselines over our two proposed datasets.

BERTSUMEXT. (Liu and Lapata, 2019) BertSumExt model is the extractive variant of BERTSUM which is the BERT Model fine-tuned on text summarization task. In this regard, BERT `[CLS]` tokens are appended to the start of each input sentence, and their associated representations are used to predict if the sentence should be included in the final summary or not.

BERTSUMABS. (Lewis et al., 2020) BERTSUMABS is the abstractive model of BERTSUM, where a Transformers-based decoder is added to the BERT Encoder.

BART. (Lewis et al., 2020) BART is a regressive autoencoder model that is pre-trained by first corrupting the text with an arbitrary nosing function, and secondly, trying to reconstruct the original input text. BART is particularly effective when fine-tuned on text generation tasks such as summarization. As BART has both encoder and decoder pre-trained, it can be perceived as an extension to general BERT models in which only encoder is pre-trained.

4.2 Dataset

We randomly split our datasets to construct training, validation, and test sets. Specifically, for TLDR9+, we use 99-0.5-0.5 split which results in 9,139,935 (train), 43,753 (validation), and 43,749 (test) instances. To split TLDRHQ, we use 95-2.5-2.5 division yielding 1,590,132 (train), 40,481 (validation), and 40,486 (test) pairs.

4.3 Training and Hyper-parameters

To train the summarization models, we utilize HuggingFace's Transformers (Wolf et al., 2020) for BART, and the open implementation [5] of BERTSUMEXT, BERTSUMABS. We use warm-up steps of 32K, and 20K for BART and BERTSUM variants, respectively. The AdamW optimizer (Loshchilov and Hutter, 2019) is used with learning rate of $3e-5$, beta parameter of 0.98, and weight decay of 0.01 for BART model. For BERTSUM variants, we use the default Adam (Kingma and Ba, 2015) optimizer with learning rates of $2e-3$ for the encoder, and $1e-2$ for the decoder as suggested by the main paper (Liu and Lapata, 2019). For all models, we use cross-entropy loss function. We train the models on 8 Nvidia Tesla V100 GPUs for 5 epochs with early stopping of the training when the validation loss does not decrease for 3 consecutive validation steps. The validation step is done

[5] https://github.com/nlpyang/PreSumm

Model	TLDR9+			TLDRHQ		
	RG-1(%)	RG-2(%)	RG-L(%)	RG-1(%)	RG-2(%)	RG-L(%)
BERTSUMEXT (Liu and Lapata, 2019)	20.94	4.98	14.48	28.40	11.35	21.38
BERTSUMABS (Liu and Lapata, 2019)	23.05	9.48	18.07	28.96	12.08	22.08
BART (Lewis et al., 2020)	23.59	9.69	18.62	32.44	14.85	27.39
ORACLE-EXT	30.26	9.74	20.60	45.29	25.47	36.86

Table 4: ROUGE (F1) results of the state-of-the-art summarization models on the test sets of the proposed TLDR summarization datasets (TLDR9+, and TLDRHQ).

every 25K training steps. To visualize and keep track of the learning process, we use Weight and Biases (Biewald, 2020) toolkit.

5 Experimental Results

Table 4 presents the performance of the state-of-the-art summarization models on our proposed datasets in terms of ROUGE-1, ROUGE-2, and ROUGE-L scores. As indicated, BART outperforms all other models across all ROUGE variants in both datasets. This is expected as BART's both encoder and decoder have been pre-trained on a large amount of unlabelled data, unlike BERTSUM variants that only have pre-trained encoders.

Comparing abstractive models with BERTSUMEXT, we observe relatively large performance gap. This might be due to the fact that TLDRs in both TLDR9+ and TLDRHQ datasets are rather abstractive than extractive as also shown in Section 3.4. Yet with the existence of such a huge gap, the ORACLE-EXT (i.e., upper bound of an extractive summarizer) scores prove that more developed extractive summarizers can perform out-of-the-box and mitigate this gap. The performance gap on TLDR9+ brings various challenges to develop summarization models that better fit on the larger dataset that include noisy data (Kumar et al., 2020). This noise might be handled via methods such as noise-aware training models (Namysl et al., 2020), while enabling the models to benefit from the large-scale TLDR9+ dataset. We leave this part for future work.

6 Analysis

To gain insights into the qualities of summarization model, we analyze the outputs generated by the models. The diagrams demonstrating n-gram abstractiveness and percentage of novel n-grams, generated by BART and BERTSUMABS, are plotted in Figure 6. As observed, BART model appears to have a similar trend to the ground truth TLDRs. On the other hand, BERTSUMABS model has increasing n-gram abstractiveness, and novel n-gram percentage with increasing n. It is also interesting that after 6-gram, BERTSUMEXT model reaches a plateau when generating novel n-grams, but we a drop after 3-grams for BART and the ground truth TLDRs. This shows that from 1-gram to 3-gram, there are increasing number of novel words appeared in the ground-truth and BART, but after that, they both tend to copy n-grams rather than generating those.

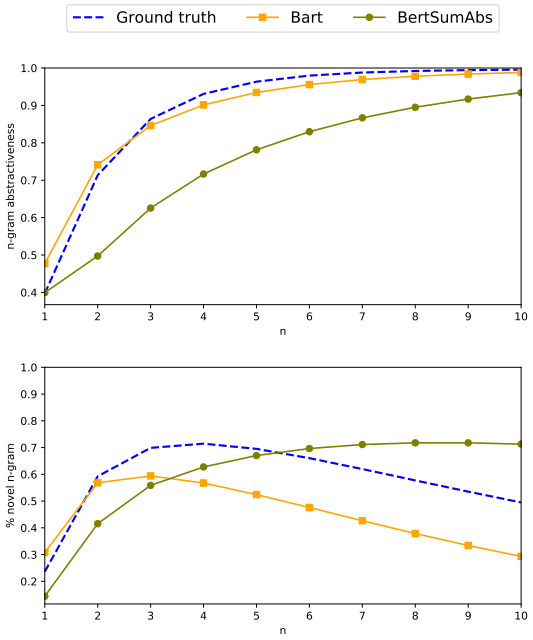


Figure 6: The n-gram abstractiveness and percentage of novel n-gram metrics across different n-grams on TLDRHQ's test set. As seen, BART generates more abstractive summaries than BERTSUMABS as it mitigates the gap between BERTSUMABS and ground truth summary.

To understand the limitation and qualities of cur-

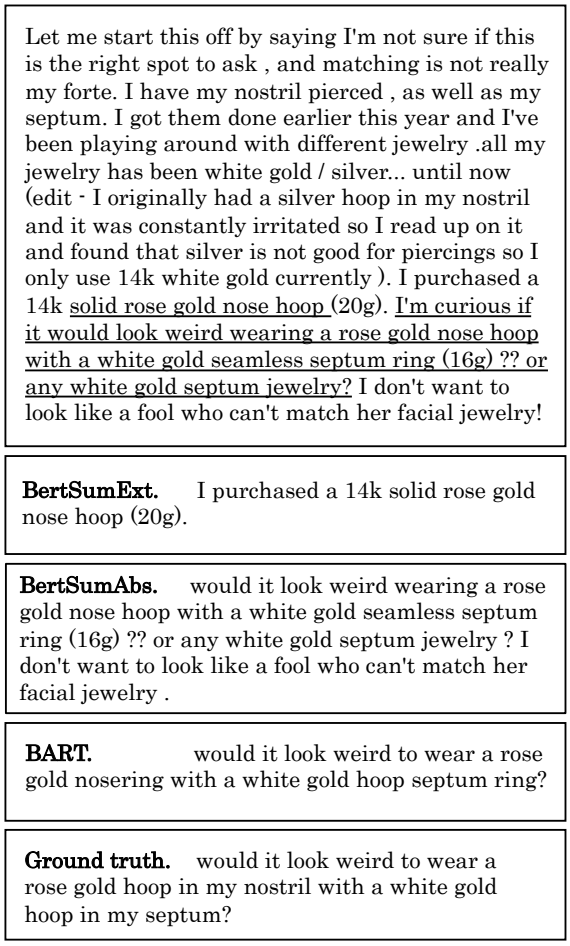


Figure 7: A sample from TLDRHQ test set along with the model generated summaries. Underlined text in source shows the important regions of the source for generating TLDR summary.

rent state-of-the-art summarization models, we conduct a qualitative analysis on several samples from TLDRHQ dataset, of which one is shown in Figure 7. Analyzing this sample, we observe that BART generated a better summary in terms of faithfulness to the ground truth TLDR. On the other hand, while BERTSUMABS could identify the important region of the source document, it has produced a longer TLDR with additional information that is present in the source, but not in the ground truth summary. BERTSUMEXT model could have identified a source sentence which is partly in connection with the ground truth TLDR, but it leaves out the most important sentence as the oracle to be extracted. Considering the upper performance of extractive summarizers (i.e., ORACLE-EXT score in Table 4), we believe that there is a large room for improvement on this dataset. Investigations of more advanced models remains for future work.

7 Conclusion

In this paper, we proposed two large-scale summarization datasets called TLDR9+, and TLDRHQ. The TLDR9+ dataset contains over 9 millions Reddit post-TLDR instances. To distill a more fine-grained dataset out of TLDR9+, we sample high-quality instances with the help of human annotations to construct TLDRHQ. Our analyses over TLDR9+ and TLDRHQ datasets show its usability for performing both extractive and abstractive summarization tasks. We further establish extractive and abstractive baseline results using state-of-the-art summarization models on both datasets. We hope our datasets can pave the path for future studies on this direction.

References

A. Althnian, D. AlSaeed, Heyam H. Al-Baity, Amani K. Samha, Alanoud Bin Dris, Najla Alzakari, A. A. Elwafa, and H. Kurdi. 2021. Impact of dataset size on classification performance: An empirical evaluation in the medical domain. *Applied Sciences*, 11:796.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and J. Blackburn. 2020. The pushshift reddit dataset. In *ICWSM*.

E. M. Bennet, R. Alpert, and A. C. Goldstein. 1954. Communications Through Limited-Response Questioning*. *Public Opinion Quarterly*, 18(3):303–308.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S. Weld. 2020. Tldr: Extreme summarization of scientific documents. In *FINDINGS*.

Sangwoo Cho, Kaiqiang Song, Chen Li, Dong Yu, H. Foroosh, and Fei Liu. 2020. Better highlighting: Creating sub-sentence summary highlights. In *EMNLP*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, W. Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *NAACL-HLT*.

Yue Dong, Yikang Shen, E. Crawford, H. V. Hoof, and J. Cheung. 2018. Banditsum: Extractive summarization as a contextual bandit. In *EMNLP*.

Sebastian Gehrmann, Y. Deng, and Alexander M. Rush. 2018. Bottom-up abstractive summarization. *EMNLP*.

Sebastian Gehrmann, Zachary M. Ziegler, and Alexander M. Rush. 2019. Generating abstractive summaries with finetuned language models. In *INLG*.

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

Matt Grenander, Yue Dong, J. C. K. Cheung, and Annie Louis. 2019. Countering the effects of lead bias in news summarization via multi-stage training and auxiliary losses. In *EMNLP*.

Max Grusky, M. Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *NAACL*.

S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Chris Kedzie, K. McKeown, and Hal Daumé. 2018. Content selection in deep learning models of summarization. In *EMNLP*.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *NAACL*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

A. Kornilova and Vladimir Eidelman. 2019. Billsum: A corpus for automatic summarization of us legislation. *ArXiv*, abs/1910.00523.

Ankit Kumar, Piyush Makhija, and Anuj Gupta. 2020. Noisy text data: Achilles' heel of bert. In *WNUT*.

Logan Lebanoff, Franck Dernoncourt, Doo Soon Kim, Lidan Wang, W. Chang, and Fei Liu. 2020. Learning to fuse sentences with transformers for summarization. In *EMNLP*.

M. Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, A. Mohamed, Omer Levy, V. Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ACL*.

Ji Liu and D. Inkpen. 2015. Estimating user location in social media with stacked denoising auto-encoders. In *VS@HLT-NAACL*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *EMNLP/IJCNLP*.

I. Loshchilov and F. Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Sean MacAvaney, Sajad Sotudeh, Arman Cohan, Nazli Goharian, Ish A. Talati, and Ross W. Filice. 2019. Ontology-aware clinical abstractive summarization. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.

Marcin Namysl, Sven Behnke, and J. Kohler. 2020. Nat: Noise-aware training for robust neural sequence labeling. In *ACL*.

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 *EMNLP*.

Shashi Narayan, Joshua Maynez, Jakub Adámek, Daniele Pighin, Blavz Bratanivc, and Ryan T. McDonald. 2020. Stepwise extractive summarization and planning with structured transformers. In *EMNLP*.

Alexander M. Rush, Harvard Seas, S. Chopra, and J. Weston. 2015. A neural attention model for sentence summarization.

A. See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL*.

Eva Sharma, Chen Li, and L. Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. In *ACL*.

Sajad Sotudeh, Arman Cohan, and Nazli Goharian. 2021. On generating extended summaries of long documents. *SDU@AAAI*, abs/2012.14136.

Sajad Sotudeh, Tong Xiang, Hao-Ren Yao, Sean MacAvaney, Eugene Yang, Nazli Goharian, and Ophir Frieder. 2020. Guir at semeval-2020 task 12: Domain-tuned contextualized models for offensive language detection. *SemEval2020*.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl;dr: Mining reddit to learn automatic summarization. In *NFiS@EMNLP*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Xue Ying. 2019. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168:022022.

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

Chenguang Zhu, Ziyi Yang, R. Gmyr, Michael Zeng, and Xuedong Huang. 2019. Make lead bias in your favor: Zero-shot abstractive news summarization. *arXiv: Computation and Language*.