SummerTime: Text Summarization Toolkit for Non-experts

Ansong Ni[†] Zhangir Azerbayev[†] Mutethia Mutuma[†] Troy Feng[†]
Yusen Zhang[♣] Tao Yu[†] Ahmed Hassan Awadallah[◇] Dragomir Radev[†]
[†]Yale University [♣]Penn State University [◇]Microsoft Research
{ansong.ni, tao.yu, dragomir.radev}@yale.edu

Abstract

Recent advances in summarization provide models that can generate summaries of higher quality. Such models now exist for a number of summarization tasks, including query-based summarization, dialogue summarization, and multi-document summarization. While such models and tasks are rapidly growing in the research field, it has also become challenging for non-experts to keep track of them. To make summarization methods more accessible to a wider audience, we develop SummerTime by rethinking the summarization task from the perspective of an NLP non-expert. SummerTime is a complete toolkit for text summarization, including various models, datasets and evaluation metrics, for a full spectrum of summarization-related tasks. SummerTime integrates with libraries designed for NLP researchers, and enables users with easy-to-use APIs. With SummerTime, users can locate pipeline solutions and search for the best model with their own data, and visualize the differences, all with a few lines of code. We also provide explanations for models and evaluation metrics to help users understand the model behaviors and select models that best suit their needs. Our library, along with a notebook demo, is available at https://github.com/Yale-LILY/SummerTime.

1 Introduction

The goal of text summarization is to generate short and fluent summaries from longer textual sources, while preserving the most salient information in them. Benefiting from recent advances of deep neural networks, in particular sequence to sequence models, with or without attention (Sutskever et al., 2014; Bahdanau et al., 2014; Vaswani et al., 2017), current state-of-the-art summarization models produce high quality summaries that can be useful in practice cases (Zhang et al., 2020a; Lewis et al., 2020). Moreover, neural summarization has broadened its scope with the introduction of more summarization tasks, such as query-based summarization (Dang, 2005; Zhong et al., 2021), long-document summarization (Cohan et al., 2018), multi-document summarization (Ganesan et al., 2010; Fabbri et al., 2019), dialogue summarization (Gliwa et al., 2019; Zhong et al., 2021). Such summarization tasks can also be from different domains (Hermann et al., 2015; Zhang et al., 2019; Cohan et al., 2018).

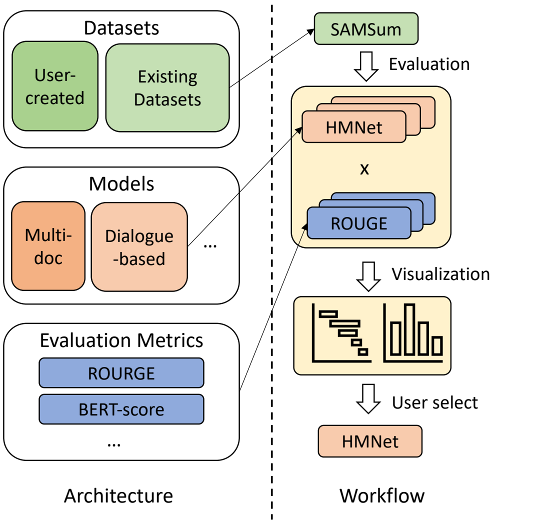


Figure 1: SummerTime is a toolkit for helping non-expert users to find the best summarization models for their own data and use cases.

However, as the field rapidly grows, it is often hard for NLP non-experts to follow all relevant new models, datasets, and evaluation metrics. Moreover, those models and datasets are often from different sources, making it a non-trivial effort for the users to directly compare the performance of such models side-by-side. This makes it hard for them to decide which models to use. The development of libraries such as *Transformers* (Wolf et al., 2020) alleviate such problems to some extent, but they only cover a narrow range of summarization models and tasks and assume certain proficiency in NLP from the users, thus the target audience is still largely the research community.

To address those challenges for non-expert users and make state-of-the-art summarizers more accessible as a tool, we introduce SummerTime, a text summarization toolkit intended for users with no NLP background. We build this library from this perspective, and provide an integration of different summarization models, datasets and evaluation metrics, all in one place. We allow the users to view a side-by-side comparison of all classic and state-of-the-art summarization models we support, on their own data and combined into pipelines that fit their own task. SummerTime also provides the functionality for automatic model selection, by constructing pipelines for specific tasks first and iteratively evaluation to find the best working solutions. Assuming no background in NLP, we list "pros and cons" for each model, and provide simple explanations for all the evaluation metrics we support. Moreover, we go beyond pure numbers and provide visualization of the performance and output of different models, to facilitate users in making decisions about which models or pipelines to finally adopt.

The purpose of SummerTime is not to replace any previous work, on the contrary, we integrate existing libraries and place them in the same framework. We provide wrappers around such libraries intended for expert users, maintaining the user-friendly and easy-to-use APIs.

2 Related Work

2.1 Text Summarization

Text summarization has been a long-standing task for natural language processing. Early systems for summarization had been focusing on extractive summarization (Mihalcea and Tarau, 2004; Erkan and Radev, 2004), by finding the most salient sentences from source documents. With the advancement of neural networks (Bahdanau et al., 2014; Sutskever et al., 2014), the task of abstractive summarization has been receiving more attention (Rush et al., 2015; Chopra et al., 2016; Nallapati et al., 2016; Celikyilmaz et al., 2018; Chen and Bansal, 2018; Lebanoff et al., 2019) while neural-based methods have also been developed for extractive summarization (Zhong et al., 2019b,a; Xu and Durrett, 2019; Cho et al., 2019; Zhong et al., 2020; Jia et al., 2020). Moreover, the field of text summarization has also been broadening into several subcategories, such as multi-document summarization (McKeown and Radev, 1995; Carbonell and Goldstein, 1998; Ganesan et al., 2010; Fabbri et al., 2019), query-based summarization (Daumé III and Marcu, 2006; Otterbacher et al., 2009; Wang et al., 2016; Litvak and Vanetik, 2017; Nema et al., 2017; Baumel et al., 2018; Kulkarni et al., 2020) and dialogue summarization (Zhong et al., 2021; Chen et al., 2021a,b; Gliwa et al., 2019; Chen and Yang, 2020; Zhu et al., 2020). The proposed tasks, along with the datasets can also be classified by domain, such as news (Hermann et al., 2015; Fabbri et al., 2019; Narayan et al., 2018), meetings (Zhong et al., 2021; Carletta et al., 2005; Janin et al., 2003), scientifc literature (Cohan et al., 2018; Yasunaga et al., 2019), and medical records (DeYoung et al., 2021; Zhang et al., 2019; Portet et al., 2009).

2.2 Existing Systems for Summarization

Transformers (Wolf et al., 2020) includes a large number of transformer-based models in its *Modelhub*[1], including BART (Lewis et al., 2020) and Pegasus (Zhang et al., 2020a), two strong neural summarizers we also use in SummerTime. It also hosts datasets for various NLP tasks in its *Datasets*[2] library. Despite the wide coverage in transformer-based models, *Transformers* do not natively support models or pipelines that can handle aforementioned subcategories of summarization tasks. Moreover, it assumes certain NLP proficiency in tis users, thus is harder for non-expert users to use. We integrate with *Transformers* and *Datasets* and import the state-of-the-art models, as well as summarization datasets into SummerTime, under the same easy-to-use framework.

Another library that we integrate with is *SummEval* (Fabbri et al., 2020), which is a collection of evaluation metrics for text summarization. SummerTime adopts a subset of such metrics in *SummEval* that are more popular and easier to understand. SummerTime also works well with *SummVis* (Vig et al., 2021), which provides an interactive way of analysing summarization results on the token-level. We also allow SummerTime to store output in a format that can be directly used by *SummVis* and its UI.

Other systems also exist for text summarization. *MEAD*[3] is a platform for multi-lingual summarization *Sumy* can produce extractive summaries from

[1] https://huggingface.co/models

[2] https://huggingface.co/datasets

[3] http://www.summarization.com/mead/

HTML pages or plain texts, using several traditional summarization methods including Mihalcea and Tarau (2004) and Erkan and Radev (2004). *OpenNMT* is mostly for machine translation, but it also hosts several summarization models such as Gehrmann et al. (2018).

3 SummerTime

The main purpose of SummerTime is to help non-expert users navigate through various summarization models, datasets and evaluation metrics, and provide simple yet comprehensive information for them to select the models that best suit their needs. Fig. 1 shows how SummerTime is split into different modules to help users achieve such goal.

We will describe in detail each component of SummerTime in the following sections. With § 3.1, we introduce the models we support in all subcategories of summarization; in § 3.2 we list all the existing datasets we support and how users can create their own evaluation set. Finally in § 3.3, we explain the evaluation metrics included with SummerTime and how they can help users find the most suitable model for their task.

3.1 Summarization Models

Here we introduce the summarization tasks SummerTime covers and the models we include to support these tasks. We first introduce the single-document summarization models (*i.e.,* "base models") included in SummerTime, and then we show how those models can be used in a pipeline with other methods to complete more complex tasks such as query-based summarization and multi-document summarization.

Single-document Summarization

The following base summarization models are used in SummerTime. They all take a single document and generate a short summary.

TextRank (Mihalcea and Tarau, 2004) is a graph-based ranking model that can be used to perform extractive summarization;

LexRank (Erkan and Radev, 2004) is also a graph-based extractive summarization model, which is originally developed for multi-document summarization, but can also be applied to a single document. It uses centrality in a graph representation of sentences to measure their relative importance;

BART (Lewis et al., 2020) is an autoencoder model trained with denoising objectives during training. This seq2seq model is constructed with a bidirectional transformer encoder and a left-to-right transformer decoder, which can be fine-tuned to perform abstractive summarization;

Pegasus (Zhang et al., 2020a) proposes a new self-supervised pretraining objective for abstractive summarization, by reconstructing the target sentence with the remaining sentences in the document, it also shows strong results in low-resource settings;

Longformer (Beltagy et al., 2020) addresses the problem of memory need for self-attention models by using a combination of sliding window attention and global attention to approximate standard self-attention. It is able to support input length of 16K tokens, a large improvement over previous transformer-based models.

Multi-document Summarization

For multi-document summarization, we adopt two popular single-document summarizers to complete the task, as this is shown to be effective in previous work (Fabbri et al., 2019).

Combine-then-summarize is a pipeline method to handle multiple source documents, where the documents are concatenated and then a single document summarizer is used to produce the summary. Note that the length of the combined documents may exceed the input length limit for typical transformer-based models;

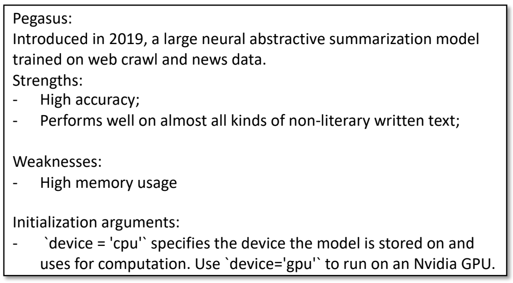
Summarize-then-combine first summarizes each source document independently, then merges the resulting summaries. Compared to the combine-then-summarize method, it is not affected by overlong inputs. However, since each document is summarized separately, the final summary may contain redundant information (Carbonell and Goldstein, 1998).

Query-based Summarization

For summarization tasks based on queries, we adopt a pipeline method and first use retrieval methods to identify salient sentences or utterances in the original document or dialogue, then generate summaries with a single-document summarization model.

TF-IDF retrieval is used in a pipeline to first retrieve the sentences that are most similar to the query based on the TF-IDF metric;

BM25 retrieval is used in the same pipeline, but BM25 is used as the similarity metric for retrieving the top-k relevant sentences.



Pegasus:
Introduced in 2019, a large neural abstractive summarization model trained on web crawl and news data.
Strengths:
- High accuracy;
- Performs well on almost all kinds of non-literary written text;

Weaknesses:
- High memory usage

Initialization arguments:
- `device = 'cpu'` specifies the device the model is stored on and uses for computation. Use `device='gpu'` to run on an Nvidia GPU.

Figure 2: A short description of the Pegasus model, SummerTime includes such short descriptions for each supported models to help user making choices.

Dialogue Summarization

Dialogue summarization is used to extract salient information from a dialogue. SummerTime includes two methods for dialogue summarization.
Flatten-then-summarize first flattens the dialogue data while preserving the speaker information, then a summarizer is used to generate the summary. Zhong et al. (2021) found that this presents a strong baseline for dialogue summarization.
HMNet (Zhu et al., 2020) explores the semantic structure of dialogues and develops a hierarchical architecture to model the long dialogue script and exploits role vectors to perform better speaker modeling.

Since we assume no NLP background of our target users, we provide a short description for every model to illustrate the strengths and weaknesses for each model. Such manually written descriptions are displayed when calling a static `get_description()` method on the model class. A sample description is shown in Fig. 2.

3.2 Datasets

With SummerTime, users can easily create or convert their own summarization datasets and evaluate all the supporting models within the framework. However, in the case that no such datasets are available, SummerTime also provides access to a list of existing summarization datasets. This way, users can select models that perform the best on one or more datasets that are similar to their task.
CNN/DM (Hermann et al., 2015) contains news articles from CNN and Daily Mail. Version 1.0.0 of it was originally developed for reading comprehension and abstractive question answering, then the extractive and abstractive summarization annotations were added in version 2.0.0 and 3.0.0, respectively;
Multi-News (Fabbri et al., 2019) is a large-scale multi-document summarization dataset which contains news articles from the site newser.com with corresponding human-written summaries. Over 1,500 sites, i.e. news sources, appear as source documents, which is higher than the other common news datasets.
SAMSum (Gliwa et al., 2019) is a dataset with chat dialogues corpus, and human-annotated abstractive summarizations. In the SAMSum corpus, each dialogue is written by one person. After collecting all the dialogues, experts write a single summary for each dialogue.
XSum (Narayan et al., 2018) is a news summarization dataset for generating a one-sentence summary aiming to answer the question “What is the article about?”. It consists of real-world articles and corresponding one-sentence summarization from British Broadcasting Corporation (BBC).
ScisummNet (Yasunaga et al., 2019) is a human-annotated dataset made for citation-aware scientific paper summarization (Scisumm). It contains over 1,000 papers in the ACL anthology network as well as their citation networks and their manually labeled summaries.
QMSum (Zhong et al., 2021) is designed for query-based multi-domain meeting summarization. It collects the meetings from AMI and ICSI dataset, as well as the committee meetings of the Welsh Parliament and Parliament of Canada. Experts manually wrote summaries for each meeting.
ArXiv (Cohan et al., 2018) is a dataset extracted from research papers for abstractive summarization of single, longer-form documents. For each research paper from arxiv.org, its abstract is used as ground-truth summaries.
SummScreen (Chen et al., 2021a) consists of community contributed transcripts of television show episodes from The TVMegaSite, Inc. (TMS) and ForeverDream (FD). The summary of each transcript is the recap from TMS, or a recap of the FD shows from Wikipedia and TVMaze.

A summary of all datasets included in SummerTime is shown as Tab. 1, it is worth noticing that the fields in this table (*i.e.,* domain, query-based, multi-doc, etc) are also incorporated in each of the dataset classes (*e.g.,* `SAMSumDataset` as class variables, so that such labels can later be used to identify applicable models. Similar with the models classes, we include a short description for each

Dataset	Domain	# Examples	Src. length	Tgt. length	Query	Multi-doc	Dialogue
CNN/DM(3.0.0)	News	300k	781	56	✗	✗	✗
Multi-News	News	56k	2.1k	263.8	✗	✓	✗
SAMSum	Open-domain	16k	94	20	✗	✗	✓
XSum	News	226k	431	23.3	✗	✗	✗
ScisummNet	Scientific articles	1k	4.7k	150	✗	✗	✗
QMSum	Meetings	1k	9.0k	69.6	✓	✗	✓
ArXiv	Scientific papers	215k	4.9k	220	✗	✗	✗
SummScreen	TV shows	26.9k	6.6k	337.4	✗	✗	✓

Table 1: The summarization datasets included in SummerTime.

of the datasets. Note that the datasets, either existing ones or user created are mainly for evaluation purposes. We leave the important task of fine-tuning the models on these datasets for future work.

3.3 Evaluation Metrics

To evaluate the performance of each supported model on certain dataset, SummerTime integrates with SummEval (Fabbri et al., 2020) and provides the following evaluation metrics for the users to understand model performance:
ROUGE (Lin, 2004) is a recall-oriented method based on overlapping n-grams, word sequences, and word pairs between the generated output and the gold summary;
BLEU (Papineni et al., 2002) measures n-gram precision and employs a penalty for brevity, BLEU is often used as an evaluation metric for machine translation;
ROUGE-WE (Ng and Abrecht, 2015) aims to go beyond surface lexical similarity and uses pretrained word embeddings to measure the similarity between different words and presents a better correlation with human judgements;
METEOR (Lavie and Agarwal, 2007) is based on word-to-word matches between generated and reference summaries, it consider two words as "aligned" based on a Porter stemmer (Porter, 2001) or synonyms in WordNet (Miller, 1995);
BERTScore (Zhang et al., 2020b) computes token-level similarity between sentences with the contextualized embeddings of each tokens.

Since assuming no NLP background from our target users, we made sure that SummerTime provides a short explanation for each evaluation metric as well as a clarification whether high or low scores are better for a given evaluation metric, to help the non-expert users understand the meaning of the metrics and use them to make decisions.

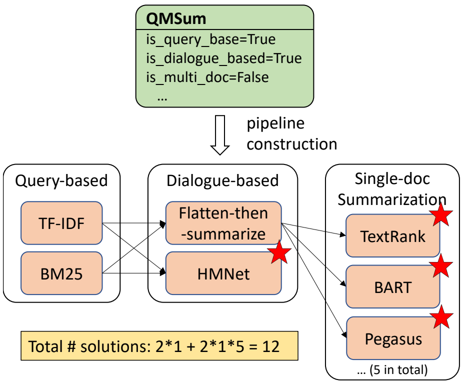


Figure 3: An illustration of how SummerTime finds solutions to a specific tasks defined by a dataset. The red star denotes that an ending point is reached.

4 Model Selection

In this section, we describe in detail about the workflow of SummerTime and how it can help our non-expert users find the best models for their use cases, which is one of the main functionalities that makes SummerTime stands out from similar libraries.

Create/select datasets The user would first either load a dataset with the APIs we provide, or choose to use one of the datasets that are already included in SummerTime. During the creation of the datasets, the users also need to specify the Boolean attributes as in Tab. 1 to facilitate next steps.

Construct pipelines After identifying the potential pipeline modules (*e.g.*, query-based module, dialogue-based module) that are applicable to the task, a combination of specific methods of such modules are put in a pool for further evaluation. An example of this process in shown in Fig. 3, SummerTime automatically constructs solutions to a specific dataset by combining the pipelines and summarization models specified in § 3.1.

Search for the best models As shown in Fig. 3, there can be a large pool of solutions to be eval-

Algorithm 1 SELECT($\mathcal{M}, \mathcal{D}, \mathcal{E}$)

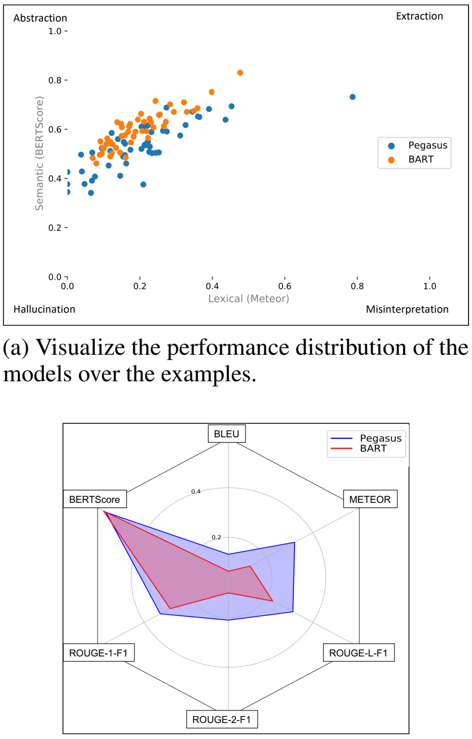
```
Input: $\mathcal{M}$: a pool of models to choose from, $\mathcal{D}$: a set of
    examples from a dataset, $\mathcal{T}$: a set of evaluation metrics,
    $d$: initial resource number, $k$: increase resource factor
Output: $M \subseteq \mathcal{M}$: a subset of models;
 1: Initialize $M = \mathcal{M}, M' = \emptyset$
 2: while $M' \neq M$ do
 3:     $D = sample(\mathcal{D}, d)$
 4:     for each $m \in M, e \in \mathcal{E}$ do
 5:         $r_m^e = eval(m, D, e)$
 6:     end for
 7:     $M' = M$
 8:     for each $m \in M$ do
 9:         if $\exists m'$ s.t. $r_{m'}^e > r_m^e, \forall e \in \mathcal{E}$ then
10:             $M = M \backslash m$
11:         end if
12:     end for
13:     $d = d * k$
14: end while
```

uated. To save time and resources in searching for best models, SummerTime adopts the idea of successive halving (Li et al., 2017; Jamieson and Talwalkar, 2016). More specifically, SummerTime first uses a small number of examples from the dataset to evaluate all the candidates and eliminate models that are surpassed by at least one other model on every evaluation metric, then it does so iteratively and gradually increases the evaluation set size to reduce the variance. As shown in Algorithm 1, the final output is a set of competing models M that are better[4] than one another on at least one metric.

Visualization In addition to showing the numerical results as tables, SummerTime also allows the users to visualize the differences between different models with different charts and *SummVis* (Vig et al., 2021). Fig. 4 shows some examples of such visualization methods SummerTime provides. A scatter plot can help the users understand the distribution of the model's performance over each example, while the radar chart is an intuitive way of comparing different models over various metrics. SummerTime can also output the generated summaries to file formats that are directly compatible with *SummVis*, so that the users can easily use it to visualize the per-instance output differences on the token level.

5 Future Work

An important piece of future work for SummerTime is to include more summarization models to enlarge the number of choices for the users, and more datasets to increase the chance of users finding similar tasks or domain for evaluation when they do not have a dataset of their own. Moreover, we would like to enable fine-tuning for a subset of smaller models we support, to enable better performance on some domains or tasks for which no pretrained models are available. We also plan to add more visualization methods for the users to better understand the differences between the outputs of various models and the behavior of each individual model itself.



(a) Visualize the performance distribution of the models over the examples.

(b) Visualize the performance of models over different evaluation metrics.

Figure 4: Examples of the visualization SummerTime provides for the users to better compare the performance between different models.

6 Conclusion

We introduce SummerTime, a text summarization toolkit designed for non-expert users. SummerTime includes various summarization datasets, models and evaluation metrics and covers a wide range of summarization tasks. It can also automatically identify the best models or pipelines for a specific dataset and task, and visualize the differences between the model outputs and performances. SummerTime is open source and available online.

[4] Note that in line 9 of the algorithm, the symbol ">" is conceptual and should be interpreted as "better than"

7 Acknowledgements

The authors would like to thank Rui Zhang, Alexander Fabbri, Chenguang Zhu, Budhaditya Deb, Asli Celikyilmaz and Rahul Jha for their advice for this work. This work is supported in part by a grant from Microsoft Research.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

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.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4106–4118, Online. Association for Computational Linguistics.

Mingda Chen, Zewei Chu, Sam Wiseman, and Kevin Gimpel. 2021a. Summscreen: A dataset for abstractive screenplay summarization. *arXiv preprint arXiv:2104.07091*.

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.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021b. Dialsumm: A real-life scenario dialogue summarization dataset. *arXiv preprint arXiv:2105.06762*.

Sangwoo Cho, Logan Lebanoff, Hassan Foroosh, and Fei Liu. 2019. Improving the similarity measure of determinantal point processes for extractive multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1027–1038.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 305–312, Sydney, Australia. Association for Computational Linguistics.

Jay DeYoung, Iz Beltagy, Madeleine van Zuylen, Bailey Kuehl, and Lucy Lu Wang. 2021. Ms2: Multi-document summarization of medical studies.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2020. Summeval: Re-evaluating summarization evaluation. *arXiv preprint arXiv:2007.12626*.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 340–348.

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.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiskỳ, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1693–1701.

Kevin Jamieson and Ameet Talwalkar. 2016. Non-stochastic best arm identification and hyperparameter optimization. In *Artificial Intelligence and Statistics*, pages 240–248. PMLR.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. 2003. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE.

Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. 2020. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631.

Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. 2020. Aquamuse: Automatically generating datasets for query-based multi-document summarization. *arXiv preprint arXiv:2010.12694*.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231.

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. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2175–2189.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2017. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Marina Litvak and Natalia Vanetik. 2017. Query-based summarization using MDL principle. In *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 22–31, Valencia, Spain. Association for Computational Linguistics.

Kathleen McKeown and Dragomir R Radev. 1995. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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*, Brussels, Belgium.

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1063–1072, Vancouver, Canada. Association for Computational Linguistics.

Jun-Ping Ng and Viktoria Abrecht. 2015. Better summarization evaluation with word embeddings for ROUGE. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, Lisbon, Portugal. Association for Computational Linguistics.

Jahna Otterbacher, Gunes Erkan, and Dragomir R Radev. 2009. Biased lexrank: Passage retrieval using random walks with question-based priors. *Information Processing & Management*, 45(1):42–54.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Martin F Porter. 2001. Snowball: A language for stemming algorithms.

François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7-8):789–816.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 27:3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Jesse Vig, Wojciech Kryscinski, Karan Goel, and Nazneen Fatema Rajani. 2021. Summvis: Interactive visual analysis of models, data, and evaluation for text summarization. *arXiv preprint arXiv:2104.07605*.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2016. A sentence compression based framework to query-focused multi-document summarization. *arXiv preprint arXiv:1606.07548*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 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.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China. Association for Computational Linguistics.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yuhao Zhang, Derek Merck, Emily Bao Tsai, Christopher D Manning, and Curtis P Langlotz. 2019. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. *arXiv preprint arXiv:1911.02541*.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6197–6208. Association for Computational Linguistics.

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 Xuanjing Huang. 2019b. A closer look at data bias in neural extractive summarization models. *EMNLP-IJCNLP 2019*, page 80.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5905–5921, Online. Association for Computational Linguistics.

Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 194–203, Online. Association for Computational Linguistics.