Extractive Summarization with Text Generator

Thang Le
VinAI Research
v.thangld16@vinai.io

Luu Anh Tuan*
Nanyang Technological University
anhtuan.luu@ntu.edu.sg

Abstract

Standard extractive systems suffer from the lack of gold training signals since existing corpora solely provide document and human-written summary pairs while disregarding extractive labels. As a result, existing methods resort to imperfect pseudo-labels that are both biased and error-prone, thereby hindering the learning process of extractive models. In contrast, text generators which are commonly employed in abstractive summarization can effortlessly overcome this predicament on account of flexible sequence-to-sequence architectures. Motivated to bypass this inherent limitation, we investigate the possibility of conducting extractive summarization with text generators. Through extensive experiments covering six summarization benchmarks, we show that high-quality extractive summaries can be assembled via approximating the outputs (abstractive summaries) of these generators. Moreover, we find that the approximate summaries correlate positively with the auxiliary summaries (i.e. a better generator enables the production of better extractive summaries). Our results signify a new paradigm for training extractive summarizers i.e. learning with generation (abstractive) objectives rather than extractive schemes.

1 Introduction

Text summarization, owing to its practical application, has received increasing interest from the research community (Nguyen and Luu, 2022, Kumar and Chakkaravarthy, 2023). Current approaches mainly follow two directions: extractive and abstractive summarization (Yadav et al., 2022). While abstractive methods skillfully paraphrase the primary contents, extractive ones are less inventive as they seek to extract salient units (e.g. sentences) without making any textual modification. Nonetheless, extractive methods effectively avoid hallucinations and inconsistencies which commonly occur in abstractive summaries (Ladhak et al., 2022). We present an illustrative example in Table 1.

Source Article
Heat the broth to the boiling point. Add your Worcestershire sauce to taste. Reduce the heat and let it sit for 3 to 5 minutes. Alternatively, you can sift flour directly into the gravy, but that won't taste as good.
Abstractive Summary
Heat the broth in a large saucepan over medium heat. Sift the flour into the gravy.
Extractive Summary
Heat the broth to the boiling point. Alternatively, you can sift flour directly into the gravy, but that won't taste as good.

Table 1: An example from the WikiHow dataset showcasing an abstractive summary (BART) and an extractive summary (our method). Here the abstractive summary hallucinates the information *boiling point* to *medium heat* while the extractive summary preserves this detail as there is no textual change.

The training of abstractive models is rather straightforward as they can fit arbitrary target sequences (Sutskever et al., 2014, Shi et al., 2021). Meanwhile, extractive models suffer from the lack of gold training labels since most existing datasets only provide document and human-written summary pairs while disregarding extractive labels (Nallapati et al., 2017). The annotation process for manually obtaining these labels is also both labor-intensive and hard to control (Cheng and Lapata, 2016, Narayan et al., 2018b), further diminishing the presence of high-quality supervision. As a result, training labels for extractive models have often been secured via heuristic algorithms (Nallapati et al., 2017, Zhou et al., 2018, Xu et al., 2020, Zhang et al., 2023) which produce suboptimal alternatives (Zhang et al., 2018) and contain labeling biases (Xing et al., 2021) that lead to un-

*Corresponding Author

derfitting (Narayan et al., 2018b) as well as the error propagation phenomenon (Xu and Lapata, 2023). Attributable to these instigating problems, research on acquiring better extractive labels has always been an actively developed topic (Jia et al., 2022, Xu and Lapata, 2023).

This gives rise to an intriguing research question: **Can we construct good extractive summarization models that learn directly from the ground truth summaries?** Being able to learn directly from the ground-truth summaries should eliminate the reliance on imperfect labeling algorithms which potentially introduce noise in the training process and allow learned models to make full use of available resources (summaries). However, deriving such a methodology is non-trivial as extractive models need to produce extract outputs (Liu and Lapata, 2019) which are often sentences or sub-sentential units (Zhou et al., 2020) that originate from the source document. Meanwhile, ground truth summaries are often abstractively written snippets that do not conform to this constraint and necessitate fine-grained token-level output modelings which aren't inherent in the decoder of extractive models (Cheng et al., 2023). In contrast, these limitations can be seamlessly overcome with abstractive models which are typically based on flexible seq2seq architectures (Nallapati et al., 2016). As summarization datasets are inherently extractive to a certain degree, abstractive models trained on these sources likely exhibit unequivocal extractive behaviors (Song et al., 2020). Previous works have characterized this property as the faithfulness-abstractiveness tradeoff (Ladhak et al., 2022) and opt to find a balance in extractivity that does not hurt the performance of abstractive models (Ge et al., 2023, Dixit et al., 2023). We hypothesize, however, that this property can serve as essential clues in transforming abstractive models into compelling extractive ones that concomitantly overcome the aforementioned gap. To decipher this conjecture as well as answer the research question, we propose to approximate the output summaries of abstractive models with heuristic algorithms, thereby deriving summaries of extractive formats. With the aim of examining the quality of these extractive outputs, we conduct exhaustive evaluations spanning six summarization benchmarks while taking into account state-of-the-art standard methods on extractive summarization. To our surprise, the evaluated models perform competitively, even outperform previous state-of-the-art methods across a wide range of settings despite not undergoing any (sentential) extractive training. Remarkably, these results are achieved without setting any extraction threshold which is unprecedented in traditional methods.

In summary, our contributions can be listed as follows:

- We present Abstract2Extract (A2E), a methodology that transforms existing abstractive models into powerful extractive epitomes by taking advantage of their innate extractiveness via heuristic algorithms, all the while not incurring additional training or inference cost.
- We demonstrate through experiments on a variety of domains that A2E models exhibit either superior or comparable performance to previous state-of-the-art extractive methods despite not undertaking any extractive supervision. In addition, A2E keeps track of both abstractive and extractive summaries which provides a straightforward unification of the two paradigms.

2 Related Works

Abstractive Summarization Together with the introduction of neural sequence-to-sequence learning (Sutskever et al., 2014), progress in the field significantly skyrocketed (Nallapati et al., 2016, Liu et al., 2022). To better guide the learning of these models and avoid hallucination, many existing works attempt to explicitly control the content selection process (Wang et al., 2020, Jiang et al., 2021, Nguyen et al., 2021b, Ladhak et al., 2022). Among different categories of guidance, extractive summarization and extractive labels have also been adopted. For example, Liu and Lapata, 2019 trained a two-stage model where the base architecture is sequentially fine-tuned on the extractive and abstractive summarization tasks. Bao and Zhang, 2021 rewrote the whole extractive summaries conditioned on the input documents. Similarly, Dou et al., 2021 designed a framework incorporating extractive guidance in abstractive models and observed increased faithfulness.

Extractive Summarization Extraction summarization has often been formulated as a sentence ranking task, where the goal is to predict the importance score of each sentence and perform selection accordingly (Gupta et al., 2014, Nallapati et al., 2017). Due to the lack of extractive labels, Nallapati et al., 2017 employs a greedy approach to collectively select a subset of sentences that maximize the ROUGE (Lin, 2004) scores, whose strategy is also re-used in follow-up works (Kedzie et al., 2018, Zhong et al., 2019). This widely adopted approach, however, generates uncalibrated label sets containing biases (Jia et al., 2022) that potentially hurt the training of extractive models and further cause underfitting (Dong et al., 2018). To tackle this problem, Xu and Lapata, 2023 proposed to integrate a pool of summary candidates to derive fine-grained soft sentence labels. The approach remains limited as these scores represent merely a portion of an intractable hypothesis space and inevitably result in inferior approximators of the true ground truth which still hinder models' learning capacities.

Concurrent to our work, Varab and Xu, 2023 proposed to employ the abstractive model BRIO (Liu et al., 2022) as the scorer in guiding summary searches and achieved encouraging extractive results. Their approach, however, relies on the coordination property (i.e. the ability to properly rank summary hypotheses) which isn't inherent in most abstractive systems, and significantly degrades when the underlying model does not possess this characteristic. In contrast, we do not make any assumption about the underlying abstractive model and solely make use of the generated outputs as pseudo-references in heuristic practices which follows a black-box manner with high flexibility. Different from theirs, our approach neither diverges from the generation process of abstractive models nor additionally incurs any inference cost and can therefore seamlessly support the creation of dual summaries (i.e. abstractive and extractive).

3 Abstract2Extract

3.1 From Generation to Extraction

Given an input document D, suppose that we have access to a sequence-to-sequence abstractive summarization model M_θ which imitates the conditional likelihood $P_\theta(Y|D) = \prod_{i=1}^{t} P_\theta(Y_t|Y_{<t}, D)$

where Y represents the output summary. This probability distribution is primarily learned via the Maximum Likelihood Estimation (MLE) objective (Rehman et al., 2023). At inference time, heuristic decoding methods (e.g. beam decoding) are customarily used to generate the output sequence Y autoregressively (Kasai et al., 2022).

Denote $Y_A = M_\theta(D)$ as the abstractive summary generated from M_θ. We opt to find an alternative extractive summary Y_E conditioned on Y_A: $Y_E = \mathrm{argmax}_{Y_E \in H(D)} Q(Y_E, Y_A)$ where $H(D)$ is the hypothesis space[1] and $Q(.)$ is the reference metric.

This formulation allows the construction of extractive summaries conditioned on the directly learned ground truth distribution Y while also taking advantage of useful fine-grained token-level output information which is otherwise impracticable in standard extractive paradigms. Accordingly, we can also bypass the problem of error propagation/noisy signal caused by imperfect pseudo-labels employed in extractive training.

3.2 Approximator

Since the pool of probable extractive candidates is literally intractable making the argmax operation expensive, we adopt heuristic practices to efficiently deduce good targets.

We delineate two groups of heuristics: *summary output* - which produces summary-level (or set-level) rankings and *sentence output* - which yields sentence-level rankings. For the prior, we choose the summary (or set) with the highest ranking as the extractive summary. For the latter, we select the top K_S highest-ranked sentences to acquire the extractive summary.

3.2.1 Summary Output

These algorithms explore the hypothesis space $H(D)$ and maintain the rankings of summaries (or sets) found during the process based on $Q(.)$. We harness two classic algorithms that are highly capable: *greedy* and *beam search*.

Greedy Search Starting from an empty selection set $H = \{\}$, at each step t, the algorithm picks the locally highest quality sentence $s_t = \mathrm{argmax}_{s_t \in H'} Q(H \cup s_t, Y_A)$ and perform update $H = H \cup s_t$, where $H' = D_S \setminus H$ and D_S is the set of input sentences. The algorithm converges when the quality of the selection set cannot be further improved i.e. $\max_{s_t \in H'} Q(H \cup s_t, Y_A) \leq Q(H, Y_A)$ or additional constraints are met (e.g. maximum search steps).

Beam Search Instead of keeping only the locally best candidate H, beam search maintains a list of K_C best found sets $\{H_i\}_{i=1..K}$. At each iteration, it sequentially expands and prunes candidates in

[1]The set of all possible extractive summaries

$\{H_i\}$ based on Q. Similar to greedy search, the algorithm converges if either no better candidate gets discovered or extra restrictions are fulfilled.

3.2.2 Sentence Output

These algorithms are oriented to bring out rankings of individual input sentences. We exploit two scoring mechanisms: *local* and *global*.

Local Scorer For each sentence $s_i \in D_S$ in the source document, we evaluate its affinity with the auxiliary reference Y_A as $r_i = Q(s_i, Y_A)$, where Q is the established criterion. The computed affinity scores $\{r_i\}$ are then applied to determine sentence rankings.

Global Scorer Inspired by Xu and Lapata, 2023, we further incorporate summary-level information into the scoring of sentences. In particular, we first utilize beam search to retrieve a pool of K_C high-quality candidates $\{H_i\}_{i=1..K}$. Afterward, we iterate through the list and for each sentence s_i^k appearing in the candidate H_k, we update its affinity score as $r_i = r_i + Q(H_k, Y_A)$. To begin with, each affinity score is initialized as $r_i = 0$ and subsequently gets revamped according to its contribution (presence) in forming high-quality summaries.

3.3 Criterion

Employed heuristics rely on the criterion Q, which should encapsulate both relevance and conciseness in grading different sentences/summaries with respect to the pseudo-reference Y_A. While embedding-based criteria depend on latent features from pre-trained language models and can therefore capture contextualized information, they are computationally too demanding. In this work, we exploit the de-facto metric ROUGE[2] (Lin, 2004) as the optimization criterion following past literature (Chen et al., 2021, Gu et al., 2022). To justify this decision, we measure the lexical overlap between the abstractive summaries (PEGASUS) and the source documents in terms of *extractive n-grams* in Table 2. Overall, we observe high overlap rates which signify the method's feasibility.

	CD	XS	RD	WH	PM	MN
Uni.	94.46	73.95	89.75	88.59	82.00	94.39
Bi.	77.80	26.03	44.41	48.85	60.77	72.59

Table 2: Percentage of *extractive (non-novel) n-grams* in PEGASUS's summaries. **CD**, **XS**, **RD**, **WH**, **PM** and **MN** stand for **CNN/DailyMail**, **XSum**, **RedditTIFU**, **WikiHow**, **PubMed** and **Multi-News**, respectively.

4 Experiments

4.1 Settings

To examine our approaches, we conduct experiments on six summarization datasets: **CNN/DailyMail** (Nallapati et al., 2016) - a news-story dataset from the *CNN* and *Daily Mail* websites; **XSum** (Narayan et al., 2018a) - an extreme summarization dataset from *BBC*; **Reddit-TIFU** (Kim et al., 2019) - a social media dataset from the *TIFU* subreddit; **WikiHow** (Koupaee and Wang, 2018) - a knowledge-based dataset from the $WikiHow$ website; **PubMed** (Cohan et al., 2018) - a medical dataset; **Multi-News** (Fabbri et al., 2019) - a multi-document news summarization dataset. [3]

As underlying abstractive systems, we primarily use the following four models: **PEGASUS** (Zhang et al., 2020a) - a transformer model pre-trained with gap-sentence objectives; **BART** (Lewis et al., 2020) - a similar architecture pre-trained with denoising objectives; **BRIO** (Liu et al., 2022) - a multi-task optimized model; **PRIMERA** (Xiao et al., 2022) - a longformer encoder-decoder model pre-trained with the pyramid framework. During inference, we use beam decoding with hyperparameters determined following respective papers[4]. To guide heuristic algorithms, we use the ROUGE-1 F1 score in all experiments unless explicitly specified otherwise[5].

4.2 Can abstractive summaries serve as good pseudo-references ?

For the first experiment, we examine the quality of the approximate summaries with respect to the abstractive pseudo-references. In particular, we show the results in Table 3. For evaluation, we use an average of the three ROUGE scores i.e. ROUGE-1, ROUGE-2 and ROUGE-L F1 scores. Column **A**, **E** and Δ each denotes scores of the abstractive, approximate extractive and the accompanying quality loss during approximation. We additionally highlight the highest score in each block (or lowest in terms of loss).

On all datasets, we observe a consistent trend that **the superior the abstractive summary, the**

[2] ROUGE only depends on lexical overlap and is therefore significantly cheaper to compute.

[3] Full statistics in Appendix A

[4] Checkpoint details in Appendix B

[5] See Section C.0.3

better the extractive summary. This indicates that if we use a better abstractive model, we can expect a higher-quality extractive summary. Moreover, the finer the abstractive summary, the higher the transfer loss. This indicates that **high-grade abstractive summaries pose increasing difficulties in approximation**. Besides, we observe that the transfer loss is typically inflated on abstractive datasets such as XSum and WikiHow. Meanwhile, on fairly extractive datasets such as CNN/DailyMail or Multi-News, the approximate extractive summaries are comparatively close in quality compared to the auxiliary summaries. Ultimately, we find that **abstractive summaries can serve as good pseudo-references**, enabling extraction of non-trivial summaries on all datasets.

Dataset	Model	A↑	E↑	Δ↓
CNN/DailyMail	PEGASUS	33.08	32.88	**0.2**
	BART	35.72	33.26	2.46
	BRIO	**38.99**	**35.31**	3.68
XSum	PEGASUS	37.03	16.71	20.33
	BART	35.16	16.60	**18.56**
	BRIO	**38.51**	**17.01**	21.5
RedditTIFU	PEGASUS	21.50	16.65	**4.85**
	BART	**22.96**	**17.71**	5.25
WikiHow	PEGASUS	34.35	24.71	**9.65**
	BART	**35.59**	**25.62**	9.98
PubMed	PEGASUS	32.78	32.02	**0.76**
	BART	33.54	32.59	0.95
	PRIMERA	**33.89**	**32.88**	1.01
Multi-News	PEGASUS	36.57	35.16	1.41
	BART	36.32	35.12	**1.2**
	PRIMERA	**38.65**	**36.73**	1.92

Table 3: Abstractive and approximate extractive summaries (greedy search).

4.3 Comparison between approximators

Dataset	Model	-SUMMARY- GREEDY	BEAM	-SENTENCE- LOCAL	GLOBAL
CNN/DailyMail	PEGASUS	32.88	**32.96**	30.53	31.20
	BART	33.26	**33.39**	30.61	31.73
	BRIO	35.31	**35.55**	31.86	33.77
XSum	PEGASUS	16.71	**16.77**	15.77	15.25
	BART	16.60	**16.64**	15.64	15.15
	BRIO	17.01	**17.11**	15.85	15.44
RedditTIFU	PEGASUS	16.65	**16.69**	16.41	16.47
	BART	17.71	**17.75**	17.38	17.24
WikiHow	PEGASUS	24.71	**24.74**	23.46	23.93
	BART	25.62	**25.63**	23.59	24.26
PubMed	PEGASUS	32.02	31.97	32.61	**33.05**
	BART	32.59	32.55	32.68	**33.15**
	PRIMERA	32.88	32.89	32.80	**33.33**
Multi-News	PEGASUS	35.16	35.10	33.68	**35.23**
	BART	**35.12**	35.11	33.34	34.80
	PRIMERA	36.73	**36.75**	34.02	36.14

Table 4: Comparison between heuristic algorithms

Next, we present a comprehensive comparison of different algorithms. For *sentence output* heuristics, we determine the optimal extraction threshold based on grid search in the range $[1..32]$ and select the top highest-scored sentences according to this threshold. We report an average of the three ROUGE variants[6] in Table 4. We highlight the best heuristic for each model, and underline the better heuristic in each category. In most cases, ***summary output* heuristics produce the best summaries**, with *beam search* typically improves over *greedy search*. For those with *sentence output*, we find that the *global scorer* often achieves better results than the *local scorer*. These observations show that summary-wise (or set-wise) comparisons are necessary to deduce good extractive summaries. Drawing on this conclusion, we focus on *summary output* heuristics for the rest of the paper.

4.4 Comparison with standard extractive methods

Model	R-1	R-2	R-L
ORACLE (upper bound)	58.67	32.26	53.96
Customized Extractive Methods			
LEAD-3 (2020)	40.43	17.62	36.67
BERTSum (2019)	42.57	19.96	39.04
MatchSum (2020)	44.41	20.86	40.55
CoLo (2022)	44.58	21.25	40.65
SetSum (2023)	44.62	20.81	40.76
DiffuSum (2023)	**44.83**	**22.56**	40.56
Abstractive-driven Methods			
BART - GenX Search (2023)	38.46	16.43	34.93
BRIO - GenX Search (2023)	43.57	20.55	40.01
PEGASUS - A2E Greedy	41.69	18.93	38.03
PEGASUS - A2E Beam	41.78	18.96	38.15
BART - A2E Greedy	42.03	19.26	38.5
BART - A2E Beam	42.00	19.32	38.67
BRIO - A2E Greedy	44.18	21.15	40.6
BRIO - A2E Beam	44.44	21.29	**40.92**

Table 5: Results on **CNN/DailyMail**.

We examine the quality of the obtained summaries with respect to standard extractive systems. For reference purposes only, we provide the **ORACLE** results which involve executing *greedy search* on the ground truth summaries, serving as the upper bound of all extractive systems. Next, we specifically consider the strong baselines: **LEAD-k** (extracting first k sentences), **BERTSum** (Liu and Lapata, 2019) - a sentence-level summarizer with BERT, **MatchSum** (Zhong et al., 2020) - a

[6] ROUGE-1, ROUGE-2 and ROUGE-L

Model	R-1	R-2	R-L
ORACLE (upper bound)	33.15	7.52	23.79
Customized Extractive Methods			
BERTSum (2019)	22.86	4.48	17.16
MatchSum (2020)	24.86	4.66	18.41
CoLo (2022)	24.51	5.04	18.21
SetSum (2023)	24.80	4.59	18.52
DiffuSum (2023)	24.00	**5.44**	18.01
Abstractive-driven Methods			
BRIO - GenX Search (2023)	17.90	2.79	13.36
PEGASUS - A2E Greedy	25.79*	5.23	19.10*
PEGASUS - A2E Beam	25.86*	5.21	19.23*
BART - A2E Greedy	25.61*	5.20	19.00*
BART - A2E Beam	25.65*	5.23	19.10*
BRIO - A2E Greedy	26.2*	5.39	19.44*
BRIO - A2E Beam	**26.31***	5.37	**19.64***

Table 6: Results on **XSum**.

Model	R-1	R-2	R-L
ORACLE (upper bound)	38.41	11.92	29.8
Customized Extractive Methods			
BERTSum (2019)	23.86	5.85	19.11
MatchSum (2020)	25.09	6.17	20.13
CoLo (2022)	25.06	5.90	19.52
SetSum (2023)	25.49	6.39	20.33
Abstractive-driven Methods			
PEGASUS - A2E Greedy	24.57	5.72	19.66
PEGASUS - A2E Beam	24.63	5.68	19.75
BART - A2E Greedy	26.1	**6.48**	20.55
BART - A2E Beam	**26.12**	6.41	**20.72**

Table 7: Results on **RedditTIFU**.

Model	R-1	R-2	R-L
ORACLE (upper bound)	45.39	13.93	41.76
Customized Extractive Methods			
BERTSum (2019)	30.31	8.71	28.24
MatchSum (2020)	31.85	8.98	29.58
SetSum (2023)	31.66	8.72	29.36
Abstractive-driven Methods			
PEGASUS - A2E Greedy	33.40*	9.72*	31.00*
PEGASUS - A2E Beam	33.44*	9.72*	31.07*
BART - A2E Greedy	34.65*	**10.05***	32.12*
BART - A2E Beam	**34.66***	10.01*	**32.22***

Table 8: Results on **WikiHow**.

Model	R-1	R-2	R-L
ORACLE (upper bound)	48.92	19.71	44.58
Customized Extractive Methods			
BERTSum (2019)	41.05	14.88	36.57
MatchSum (2020)	41.21	14.91	36.75
SetSum (2023)	41.53	15.11	36.88
DiffuSum (2023)	41.40	15.55	37.48
CoLo (2022)	41.93	16.51	38.28
MemSum (2022)	**43.08**	16.71	38.30
Abstractive-driven Methods			
PEGASUS - A2E Greedy	41.65	16.25	38.15
PEGASUS - A2E Beam	41.59	16.22	38.11
BART - A2E Greedy	42.37	16.54	38.85*
BART - A2E Beam	42.32	16.51	38.82*
PRIMERA - A2E Greedy	42.72	16.76	39.16*
PRIMERA - A2E Beam	42.71	**16.77**	**39.18***

Table 9: Results on **PubMed**.

Model	R-1	R-2	R-L
ORACLE (upper bound)	62.77	30.47	57.64
Customized Extractive Methods			
BERTSum (2019)	45.80	16.42	41.53
MatchSum (2020)	46.20	16.51	41.89
SetSum (2023)	46.33	16.80	42.00
Abstractive-driven Methods			
PEGASUS - A2E Greedy	45.99	17.4*	42.1
PEGASUS - A2E Beam	45.86	17.39*	42.05
BART - A2E Greedy	46.21	16.84	42.32*
BART - A2E Beam	46.17	16.84	42.32*
PRIMERA - A2E Greedy	**47.71***	18.67*	43.81*
PRIMERA - A2E Beam	**47.71***	**18.69***	**43.86***

Table 10: Results on **Multi-News**.

two-stage matching framework, **CoLo** (An et al., 2022) - an one-stage re-ranking framework, **SetSum** (Cheng et al., 2023) - a set prediction network, **DiffuSum** (Zhang et al., 2023) - a transformer-based denoising diffusion framework, **MemSum** (Gu et al., 2022) - a highly customized model for long extractive summarization. We also provide comparisons with **GenX** (Varab and Xu, 2023) - a concurrent work close to ours that also employs abstractive model but relies on likelihood comparison instead of pseudo-references. We compare these standard methods with *summary output* heuristics. In addition, we **do not set any extraction threshold** for these heuristics (*greedy* and *beam search*) i.e. the algorithms converge only when no better candidates are found without extra constraints such as a maximum number of extracted sentences or search steps. Also, we use a default beam width of 4 unless specified otherwise[7]. For evaluation, we report the ROUGE-1, ROUGE-2 and ROUGE-L F1 scores achieved with each system. The results are presented in Table 5, 6, 7, 8, 9 and 10[8].

On CNN/DailyMail, our methods coupled with the BRIO model achieve results on par with state-of-the-art models such as MatchSum and DiffuSum. The PEGASUS/BART models also perform comparably to the BERTSum baseline. Noticeably, the BRIO - A2E Beam model achieves the high-

[7] See Section C.0.4

[8] We embolden the highest value and use asterisk "*" to denote results that significantly improve over the best baseline as measured via bootstrap testing with 95% confidence interval.

est ROUGE-L score. Compared with GenX, we also achieve consistently better scores. In addition, when the underlying system is not coordinated, our models do not significantly degrade, unlike GenX. For example, when switching from BRIO to BART whose summaries are of lower quality, we only suffer a 2-point drop in ROUGE-1 compared to GenX which degenerates by 5 ROUGE-1 points.

On XSum, our models consistently produce summaries with higher quality than baseline methods, especially in terms of ROUGE-1 and ROUGE-L.

On RedditTIFU and WikiHow, our models also outperform existing systems. In particular, our BART - A2E models surpass the best baseline SetSum on RedditTIFU. On WikiHow, our advantages are even more amplified, as all models improve 2 to nearly 3 ROUGE-1 points over the state-of-the-art model MatchSum with similar gains in ROUGE-2 and ROUGE-L.

On PubMed and Multi-News, we continually set new state-of-the-arts with persistent advances. On PubMed, our least competent models (PEGASUS) perform better than most previous systems while our best models (PRIMERA) outperform the best baseline MemSum regarding ROUGE-2 and ROUGE-L. We also observe similar results on Multi-News where our PEGASUS/BART models exceed most baselines and our PRIMERA models achieve absolute improvement over all methods.

Conclusively, we reach new **state-of-the-arts in extractive summarization** despite not undergoing customized training.

4.5 Evaluation with other metrics

We additionally report the results in terms of SummaQA (Scialom et al., 2019) and BERTScore (Zhang et al., 2020b). The prior is based on a question answering framework whereas the latter relies on greedy matching of contextualized embeddings. We repeat the comparisons with the MatchSum system. For generators, we use BRIO on CNN/DailyMail & XSum, BART on RedditTIFU & WikiHow and PRIMERA on PubMed & Multi-News. As for heuristics, we simply use *greedy search*. Dataset names are abbreviated[9]. We report the results in Table 11 and 12. Aligning with the previous section, we achieve consistently superior results on all benchmarks.

[9]Abbreviation follows Table 2

	CD	XS	RD	WH	PM	MN
MatchSum	25.96	9.88	2.25	2.19	2.75	8.04
Our method	**27.15**	**11.92**	**2.58**	**3.59**	**3.09**	**9.74**

Table 11: Results in SummaQA scores.

	CD	XS	RD	WH	PM	MN
MatchSum	64.05	57.24	52.55	56.29	58.83	61.00
Our method	**65.11**	**58.84**	**54.49**	**58.07**	**60.54**	**62.88**

Table 12: Results in BERTScore scores.

4.6 Manual Evaluation

To examine whether the automated evaluations align with human preferences, we further conduct a manual evaluation campaign. In particular, we randomly sampled 150 instances from the CNN/DailyMall test set and included extracted summaries from the MatchSum system and the A2E Greedy - BRIO model (we avoided samples where both extracted sentence sets are identical). Following Cheng et al., 2023, we invited three volunteers who are professional English speakers to examine the summaries based on two criteria: informativeness and coherence. System outputs were presented in random order and no participant was aware of the different systems beforehand. Each participant then, given the source article and gold reference, elected the summary which he/she preferred for each aspect separately. Each system then received one point for every vote.

We present the average results (percentage) in Table 13. It is clear that the summaries produced by A2E were preferred more by humans on both categories. Moreover, we obtained these results with substantial inter-annotator agreement as indicated by Fleiss' Kappa scores (Fleiss, 1971), which we show in Table 14.

	Informativeness	Coherence
MatchSum	19.56	24.67
Our method	**80.44**	**75.33**

Table 13: Human evaluation results on CNN/DailyMall.

	Informativeness	Coherence
Fleiss' Kappa	0.7034	0.6532

Table 14: Inter-Annotator Agreement.

4.7 Analysis on Lead Bias

Traditionally extractive systems often exhibit spurious correlations with beginning sentence positions, also known as *lead bias*, which emerges from an imbalance in the distribution of information positioning (Grenander et al., 2019, Xing et al., 2021). Compared to previous approaches, in our method, the learning process is identical to abstractive generation and the model thus learns to actually generate summaries rather than simply extract sentences which should supposedly lessen this spurious correlation.

To verify this argument, we examine the positions of sentences extracted with our models and the MatchSum system. In particular, we report the percentage of sentences with relative positions belonging to each of the range $0-10\%$, $10-30\%$ and $30-100\%$. We experiment with CNN/DailyMall - a dataset where lead bias is prevalent (See et al., 2017), and report results in Table 15:

	$0-10\%$	$10-30\%$	$30-100\%$
MatchSum	39.17	43.11	17.72
A2E Greedy - BRIO	**31.19**	**37.57**	**31.24**
A2E Greedy - BART	**27.01**	**39.97**	**33.02**

Table 15: Distribution of sentence positions in CNN/DailyMall extractive summaries.

As we expected, A2E models suffer less from lead bias. However, we find that the bias still exists. Specifically, when we compared the sentence positions of A2E models that were trained in-domain on XSum - a dataset with weak lead bias (Narayan et al., 2018a), versus cross-domain from CNN/DailyMall, we observed higher rates of extraction in the beginning parts for the latter. We illustrate this in Table 16.

	$0-10\%$ (In)	$0-10\%$ (Cross)
A2E Greedy - BRIO	**8.11**	25.75
A2E Greedy - BART	**8.65**	26.00

Table 16: Propagation of dataset bias on information positioning. Models were tested either in-domain or cross-domain (from CNN/DailyMall) on the XSum dataset.

This means that completely eliminating lead bias remains a non-trivial feat, which aligns with the results from Xing et al., 2021.

4.8 Further Optimization

We next study whether exact optimization can yield better extractive summaries (than heuristics). To experiment with this direction, we sample 100 documents from the CNN/DailyMail test set, each containing 9 sentences. We then compare the quality of extractive summaries conditioned on the abstractive ones (BRIO) obtained through *greedy search* and *brute force*[10]. We show the results in Table 17. Even though the gains are visible, the speed trade-offs are enormous.

	R-1	R-2	R-L	Speed
Greedy	47.2	24.65	42.95	**270.6** (iter/s)
Brute Force	**47.58**	**24.91**	**43.43**	4.6 (iter/s)

Table 17: Results with greedy search and brute force on CNN/DailyMall.

4.9 Cross-domain generalization

Although abstractive models are known to possess certain generalization capabilities (Chen et al., 2020), whether our approaches can leverage these properties remains a puzzle. To elucidate this matter, we employ a BRIO model fine-tuned on the CNN/DailyMail dataset and conduct cross-dataset inference on three benchmarks with distinct properties: XSum, RedditTIFU and WikiHow. We also compare with standard systems such as BERTSum, MatchSum and additionally include results for GenX. As Xu and Lapata, 2023 use ROUGE-L when reporting performances of standard systems, we also report ROUGE-L scores for our models accordingly. We show the results in Table 18. It can be inferred that not only can our models **generalize across domains** but we also achieve massive improvements especially when testing on non-news domain such as RedditTIFU and WikiHow.

Model	XS	RD	WH
Customized Extractive Methods			
BERTSum (2019, 2023)	15.62	17.06	25.39
MatchSum (2020, 2023)	15.75	17.82	25.1
Abstractive-driven Methods			
BRIO - GenX Search (2023)	15.92	-	-
BRIO - A2E Greedy	15.96	19.25*	27.02*
BRIO - A2E Beam	**16.00**	**19.51***	**27.06***

Table 18: Results for cross-domain summarization (ROUGE-L). Models are trained on the CNN/DailyMail dataset.

[10]Equivalent to conducting the argmax operation in Section 3.1

4.10 Faithfulness

In Section 4.2, we observed that the extractive summaries yielded lower ROUGE scores than their abstractive counterparts. However, **are extractive summaries actually inferior** ? We re-evaluate the two types of summaries from a distinct but important aspect - *faithfulness*. In particular, we collect the PEGASUS model's summaries along with the extractive ones obtained via *greedy search* and feed them through SummaC-Conv (Laban et al., 2022) - a strong factuality metric. We report the results in Table 19. As we can see, the extractive summaries are far more faithful than the abstractive ones, making them more reliable in real world deployment. Nevertheless, our methods always keep track of the extractive summaries along with the abstractive ones which allows the end users to freely choose whichever kind that suits their needs.

	CD	XS	RD	WH	PM	MN
Abstractive	51.96	24.97	28.32	68.02	47.21	62.12
Our method	**90.82**	**90.19**	**91.25**	**88.87**	**86.9**	**91.52**

Table 19: Faithfulness evaluation with abstractive summaries (PEGASUS) and extractive summaries (our method).

4.11 Application in hallucination detection

Unlike extractive systems, abstractive ones are more prone to factual errors (Cao et al., 2022). Towards mitigating this phenomenon, hallucination detection models have been developed aiming to automatically detect these errors, often via comparison between the produced summary and the source document (Goyal and Durrett, 2021, Fabbri et al., 2022). However, not all information present in the source document is relevant, and thus effective, in detecting factual errors. Therefore, instead of conducting comparison with the whole document, only using a subset of the most relevant parts can possibly help in improving the performance of these systems. Accordingly, we conduct trial experiments on the AggreFact-CNN and AggreFact-XSum datasets (Tang et al., 2023), focusing on the FTSOTA split as advised in the original paper. These datasets come with prepared outputs of abstractive systems and the corresponding source articles. For each sample, similar to previous experiments, we apply *summary output* heuristics to obtain the extractive summaries and then conduct hallucination detection conditioned on these summaries along with the abstractive ones. We choose SummaC-ZS (Laban et al., 2022) as the underlying detector - a zero-shot method that's sensitive to outliers and extrema. For evaluation, we use balanced accuracy and AUC scores. Similar to Tang et al., 2023, we choose the prediction threshold based on validation performance. The results are presented in Table 20 and 21. Generally, we obtain promising improvement on both datasets. On the CNN split, the **AUC scores significantly improve** upon the original model, whereas on the XSum split, we observe **consistent gains on both metrics**. These results show that our methods can also help **develop better hallucination detectors**.

	AggreFact-CNN	
	Acc.	AUC
SummaC-ZS	64.01	0.6421
SummaC-ZS + A2E Greedy	63.88	**0.6728**
SummaC-ZS + A2E Beam (k=2)	**64.55**	0.6688
SummaC-ZS + A2E Beam (k=4)	63.88	0.6687

Table 20: Results for hallucination detection on **AggreFact-CNN (FTSOTA)**.

	AggreFact-XSum	
	Acc.	AUC
SummaC-ZS	56.35	0.5228
SummaC-ZS + A2E Greedy	57.21	0.5287
SummaC-ZS + A2E Beam (k=2)	57.58	0.5293
SummaC-ZS + A2E Beam (k=4)	**58.27**	**0.5402**

Table 21: Results for hallucination detection on **AggreFact-XSum (FTSOTA)**.

5 Conclusion

In this work, we explore the use of existing abstractive models for extractive summarization. We make no assumption on the underlying abstractive models and follow a black-box approach. Utilising abstractive summaries, we show that state-of-the-art extractive summaries can be achieved without extractive training. To validate the method's effectiveness, we conduct extensive experiments on six datasets and provide comparison with existing methods, where our models demonstrate either superior or comparable performance.

Limitations

Our works build on top of text generators (or abstractive summarizers) and thus the effectiveness of the whole pipeline also depends on these models. As we have illustrated in the experiments, a worse generator will produce auxiliary summaries with lower qualities which negatively affect the approximate summaries. Hence, adapting the methods to situations where generation models struggle to maintain peak performance (e.g. zero-shot cross-lingual (Vu et al., 2022), dialectal scenarios (Ziems et al., 2023, Le and Luu, 2023) and continual learning (Qin et al., 2023, Zhang et al., 2022, Nguyen et al., 2023)) is a worth-exploring direction. In addition, since we center on extractive summarization, the end summaries also inherit intrinsic limitations (e.g. lack of expressiveness, possible coreference issues). Nevertheless, as the pipeline seamlessly enables creation of dual summaries (i.e. abstractive and extractive), prospective future works can take advantage of this property to efficiently overcome these restrictions. For example, an end user might want an expressive summary (e.g. entertainment purposes) and accordingly choose the abstractive summary instead of the extractive one - which our method supports out of the box. Alternatively, another user might prioritize reliability (e.g. medical domains) and thus opts for the extractive summary.

Acknowledgement

We thank the anonymous reviewers and meta reviewer for their constructive feedback and helpful suggestions.

References

Chenxin An, Ming Zhong, Zhiyong Wu, Qin Zhu, Xuanjing Huang, and Xipeng Qiu. 2022. Colo: A contrastive learning based re-ranking framework for one-stage summarization. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 5783–5793. International Committee on Computational Linguistics.

Guangsheng Bao and Yue Zhang. 2021. Contextualized rewriting for text summarization. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12544–12553. AAAI Press.

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2022. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3340–3354. Association for Computational Linguistics.

William Chen, Kensal Ramos, and Kalyan Naidu Mullaguri. 2021. Genetic algorithms for extractive summarization. *CoRR*, abs/2105.02365.

Yiran Chen, Pengfei Liu, Ming Zhong, Zi-Yi Dou, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. CDEvalSumm: An empirical study of cross-dataset evaluation for neural summarization systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3679–3691, Online. Association for Computational Linguistics.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Xiaoxia Cheng, Yongliang Shen, and Weiming Lu. 2023. A set prediction network for extractive summarization. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4766–4777. Association for Computational Linguistics.

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, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 615–621. Association for Computational Linguistics.

Tanay Dixit, Fei Wang, and Muhao Chen. 2023. Improving factuality of abstractive summarization without sacrificing summary quality. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 902–913. Association for Computational Linguistics.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. Banditsum: Extractive summarization as a contextual bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3739–3748. Association for Computational Linguistics.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. Gsum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4830–4842. Association for Computational Linguistics.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1074–1084. Association for Computational Linguistics.

Alexander R. Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Qafacteval: Improved qa-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 2587–2601. Association for Computational Linguistics.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382.

Yubin Ge, Sullam Jeoung, Ly Dinh, and Jana Diesner. 2023. Detection and mitigation of the negative impact of dataset extractivity on abstractive summarization. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13963–13976. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1449–1462. Association for Computational Linguistics.

Matt Grenander, Yue Dong, Jackie Chi Kit Cheung, and Annie Louis. 2019. Countering the effects of lead bias in news summarization via multi-stage training and auxiliary losses. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6018–6023. Association for Computational Linguistics.

Nianlong Gu, Elliott Ash, and Richard H. R. Hahnloser. 2022. Memsum: Extractive summarization of long documents using multi-step episodic markov decision processes. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6507–6522. Association for Computational Linguistics.

Anand Gupta, Manpreet Kaur, Shachar Mirkin, Adarsh Singh, and Aseem Goyal. 2014. Text summarization through entailment-based minimum vertex cover. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics, *SEM@COLING 2014, August 23-24, 2014, Dublin, Ireland*, pages 75–80. The *SEM 2014 Organizing Committee.

Ruipeng Jia, Xingxing Zhang, Yanan Cao, Zheng Lin, Shi Wang, and Furu Wei. 2022. Neural label search for zero-shot multi-lingual extractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 561–570. Association for Computational Linguistics.

Yichen Jiang, Asli Celikyilmaz, Paul Smolensky, Paul Soulos, Sudha Rao, Hamid Palangi, Roland Fernandez, Caitlin Smith, Mohit Bansal, and Jianfeng Gao. 2021. Enriching transformers with structured tensor-product representations for abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 4780–4793. Association for Computational Linguistics.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Dragomir R. Radev, Yejin Choi, and Noah A. Smith. 2022. Beam decoding with controlled patience. *CoRR*, abs/2204.05424.

Chris Kedzie, Kathleen R. McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1818–1828. Association for Computational Linguistics.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2519–2531. Association for Computational Linguistics.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *CoRR*, abs/1810.09305.

G. Senthil Kumar and Midhun Chakkaravarthy. 2023. A survey on recent text summarization techniques. In *Multi-disciplinary Trends in Artificial Intelligence - 16th International Conference, MIWAI 2023, Hyderabad, India, July 21-22, 2023, Proceedings*, volume 14078 of *Lecture Notes in Computer Science*, pages 496–502. Springer.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Trans. Assoc. Comput. Linguistics*, 10:163–177.

Faisal Ladhak, Esin Durmus, He He, Claire Cardie, and Kathleen R. McKeown. 2022. Faithful or extractive? on mitigating the faithfulness-abstractiveness trade-off in abstractive summarization. In *Proceedings of* *the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1410–1421. Association for Computational Linguistics.

Thang Le and Anh Luu. 2023. A parallel corpus for Vietnamese central-northern dialect text transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13839–13855, Singapore. Association for Computational Linguistics.

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3728–3738. Association for Computational Linguistics.

Yixin Liu, Pengfei Liu, Dragomir R. Radev, and Graham Neubig. 2022. BRIO: bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2890–2903. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3075–3081. AAAI Press.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. 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, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. 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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1747–1759. Association for Computational Linguistics.

Huy Nguyen, Chien Nguyen, Linh Ngo, Anh Luu, and Thien Nguyen. 2023. A spectral viewpoint on continual relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9621–9629, Singapore. Association for Computational Linguistics.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021a. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

Thong Nguyen, Anh Tuan Luu, Truc Lu, and Tho Quan. 2021b. Enriching and controlling global semantics for text summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9443–9456.

Thong Thanh Nguyen and Anh Tuan Luu. 2022. Improving neural cross-lingual abstractive summarization via employing optimal transport distance for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11103–11111.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *ArXiv*, abs/1912.01703.

Chengwei Qin, Chen Chen, and Shafiq Joty. 2023. Lifelong sequence generation with dynamic module expansion and adaptation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6701–6714, Singapore. Association for Computational Linguistics.

Tohida Rehman, Suchandan Das, Debarshi Kumar Sanyal, and Samiran Chattopadhyay. 2023. An analysis of abstractive text summarization using pretrained models. *CoRR*, abs/2303.12796.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? In *Thirty-seventh Conference on Neural Information Processing Systems*.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3244–3254. Association for Computational Linguistics.

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, and Chandan K. Reddy. 2021. Neural abstractive text summarization with sequence-to-sequence models. *Trans. Data Sci.*, 2(1):1:1–1:37.

Kaiqiang Song, Bingqing Wang, Zhe Feng, Ren Liu, and Fei Liu. 2020. Controlling the amount of verbatim copying in abstractive summarization. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8902–8909. AAAI Press.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Liyan Tang, Tanya Goyal, Alexander R. Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin F. Rousseau, and Greg Durrett. 2023. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11626–11644. Association for Computational Linguistics.

Daniel Varab and Yumo Xu. 2023. Abstractive summarizers are excellent extractive summarizers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 330–339. Association for Computational Linguistics.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9279–9300, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhengjue Wang, Zhibin Duan, Hao Zhang, Chaojie Wang, Long Tian, Bo Chen, and Mingyuan Zhou. 2020. Friendly topic assistant for transformer based abstractive summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 485–497. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. 2022. PRIMERA: pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5245–5263. Association for Computational Linguistics.

Linzi Xing, Wen Xiao, and Giuseppe Carenini. 2021. Demoting the lead bias in news summarization via alternating adversarial learning. In *Proceedings of the*

59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021, pages 948–954. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5021–5031. Association for Computational Linguistics.

Yumo Xu and Mirella Lapata. 2023. Text summarization with oracle expectation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Divakar Yadav, Rishabh Katna, Arun Kumar Yadav, and Jorge Morato. 2022. Feature based automatic text summarization methods: A comprehensive state-of-the-art survey. *IEEE Access*, 10:133981–134003.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023. Diffusum: Generation enhanced extractive summarization with diffusion. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13089–13100. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2024. Benchmarking Large Language Models for News Summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural latent extractive document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 779–784. Association for Computational Linguistics.

Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2022. Continual sequence generation with adaptive compositional modules. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3653–3667, Dublin, Ireland. Association for Computational Linguistics.

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 Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1049–1058. Association for Computational Linguistics.

Qingyu Zhou, Furu Wei, and Ming Zhou. 2020. At which level should we extract? an empirical analysis on extractive document summarization. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 5617–5628. International Committee on 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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 654–663. Association for Computational Linguistics.

Caleb Ziems, William Held, Jingfeng Yang, Jwala Dhamala, Rahul Gupta, and Diyi Yang. 2023. Multi-VALUE: A framework for cross-dialectal English NLP. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 744–768, Toronto, Canada. Association for Computational Linguistics.

A Dataset Statistics

Statistics of the used datasets can be found in Table 22.

The data files for CNN/DailyMall[11] (Nallapati et al., 2016), XSum[12] (Narayan et al., 2018a) and Multi-News[13] (Fabbri et al., 2019) are available in Hugging Face (Lhoest et al., 2021). WikiHow (Koupaee and Wang, 2018) can be obtained via following instructions in the authors' repository[14]. For RedditTIFU (Kim et al., 2019) where there is no official split, we adopt the partitions used by Zhong et al., 2020. For PubMed (Cohan et al., 2018), we use the truncated version similar to Zhong et al., 2020 and follow-up works (An et al., 2022, Zhang et al., 2023, Cheng et al., 2023). The data files for these two datasets can be retrieved from the repository of Zhong et al., 2020[15]. For sentence segmentation, we utilize the Trankit package (Nguyen et al., 2021a).

CNN/DailyMail (Nallapati et al., 2016), XSum (Narayan et al., 2018a) and RedditTIFU (Kim et al., 2019) are available under the MIT license. WikiHow (Koupaee and Wang, 2018) and PubMed (Cohan et al., 2018) are released under the Creative Commons License (CC-BY-NC-SA). Multi-News (Fabbri et al., 2019) is provided under a Dataset Usage Agreement with LILY LAB[16].

B Implementation Details

All experiments were implemented with the PyTorch framework (Paszke et al., 2019) and the Transformers library (Wolf et al., 2019). For ROUGE calculation, we use the package *rouge-score*[17] following Gu et al., 2022. For BERTScore, we use the *microsoft/deberta-large-mnli* model as advised by the authors[18].

Our works build on text generation models and we re-use pre-trained checkpoints whenever possible. Specifically, the details are shown in Table 23. The asterisk symbol "*" implies that we fine-tune from the corresponding raw checkpoint. In particular, we use a learning rate of $1e-5$ with AdamW (Loshchilov and Hutter, 2017) optimizer and a linear decay scheduler. Every model was trained with the MLE objective for a maximum of 300K steps on an A100 GPU and the checkpoint with the lowest validation loss was selected for inference. We also include the thresholds used in experiments with *sentence output* heuristics (#Ext-Local and #Ext-Global). Additionally, the hyperparameters for generation are presented in Table 24. No tri-grams could appear more than once during the generation process.

[11] https://huggingface.co/datasets/cnn_dailymail
[12] https://huggingface.co/datasets/EdinburghNLP/xsum
[13] https://huggingface.co/datasets/multi_news
[14] https://github.com/mahnazkoupaee/WikiHow-Dataset
[15] https://github.com/maszhongming/MatchSum
[16] https://github.com/Alex-Fabbri/Multi-News/blob/master/LICENSE.txt
[17] https://pypi.org/project/rouge-score
[18] https://github.com/Tiiiger/bert_score

C Additional Ablations & Analyses

C.0.1 #Extracted Sentences

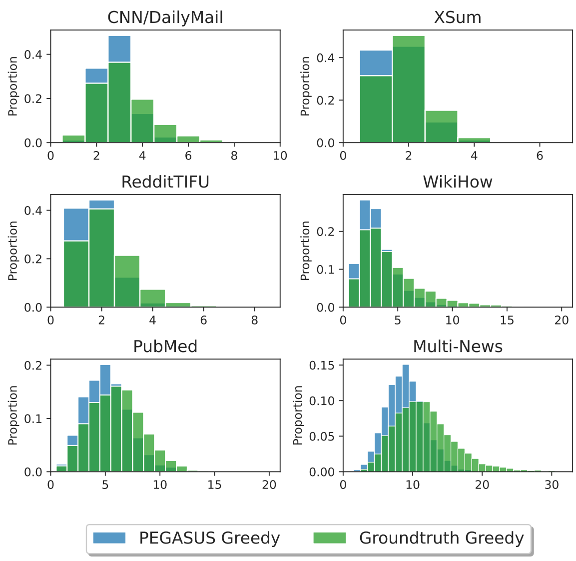

Figure 1: Length Distribution (#Sentences) - PEGASUS - A2E Greedy

We examine the length distribution of A2E models when conditioned on PEGASUS's summaries versus ground truth summaries. We present the histograms in Figure 1. It can be inferred that for the same dataset, the optimal extraction threshold differs per sample basis as indicated by the ground truth A2E outputs. Compared with the ground truth summaries, auxiliary summaries also provide good supervision imitating this property, as we can easily observe the two distributions closely resemble each other. As a result, heuristics with flexible extraction threshold (*summary output*) would gain advantages over fixed counterparts (*sentence output*).

Dataset	Source	Type	Train	Val	Test	#Tokens (doc)	#Tokens (sum)
CNN/DailyMail	News	SDS	286,010	13,295	11,490	861.5	62.5
XSum	News	SDS	203,509	11,296	11,334	469.0	26.1
RedditTIFU	Social Media	SDS	41,675	645	645	470.4	25.1
WikiHow	Knowledge Base	SDS	168,127	6,000	6,000	634.9	74.7
PubMed	Scientific Paper	SDS	83,233	4,676	5,025	561.0	260.7
Multi-News	News	MDS	44,972	5,622	5,622	921.9	277.8

Table 22: Dataset Statistics. Average sequence length was computed with BART's tokenizer.

Dataset	Model	Pre-trained	#Ext-Local	#Ext-Global
CNN/DailyMail	PEGASUS	*google/pegasus-cnn_dailymail*	3	4
	BART	*facebook/bart-large-cnn*	3	4
	BRIO	*Yale-LILY/brio-cnndm-cased*	3	3
XSum	PEGASUS	*google/pegasus-xsum*	2	3
	BART	*facebook/bart-large-xsum*	2	3
	BRIO	*Yale-LILY/brio-xsum-cased*	2	3
RedditTIFU	PEGASUS*	*google/pegasus-large*	2	3
	BART*	*facebook/bart-large*	2	3
WikiHow	PEGASUS*	*google/pegasus-large*	3	5
	BART*	*facebook/bart-large*	3	5
PubMed	PEGASUS*	*google/pegasus-large*	7	8
	BART*	*facebook/bart-large*	7	8
	PRIMERA*	*allenai/PRIMERA*	7	8
Multi-News	PEGASUS	*google/pegasus-multi_news*	8	13
	BART*	*facebook/bart-large*	8	12
	PRIMERA	*allenai/PRIMERA-multinews*	8	12

Table 23: Pre-trained Models and Extraction Threshold (*Sentence Output*). Asterisk symbol "*" indicates that we fine-tuned from the corresponding raw checkpoint.

Dataset	Model	Beam Size	Min Length	Max Length	Length Penalty
CNN/DailyMail	PEGASUS	3	56	142	0.8
	BART	2	56	142	0.8
	BRIO	128	56	142	0.8
XSum	PEGASUS	6	11	62	0.6
	BART	6	11	62	0.6
	BRIO	64	11	62	0.6
RedditTIFU	PEGASUS	1	-	128	0.6
	BART	1	-	128	0.6
WikiHow	PEGASUS	8	-	256	0.6
	BART	4	-	256	0.6
PubMed	PEGASUS	3	-	512	0.8
	BART	3	-	512	0.8
	PRIMERA	3	-	512	0.8
Multi-News	PEGASUS	8	32	256	0.8
	BART	2	32	256	0.8
	PRIMERA	5	-	1024	1.0

Table 24: Generation hyperparameters.

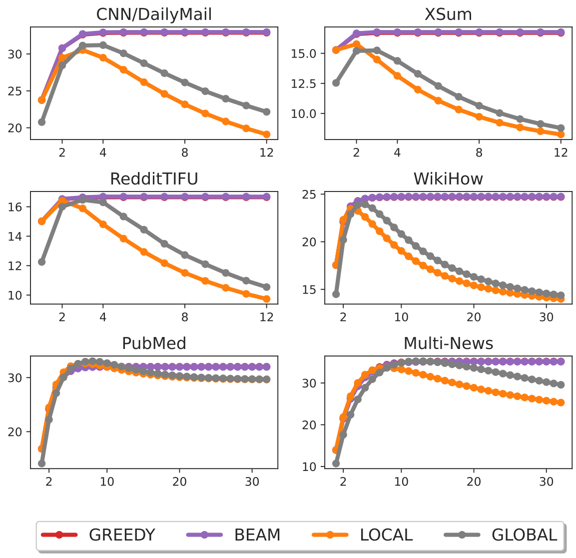

Figure 2: A2E with constrained length (*Summary Output*) and fixed threshold (*Sentence Output*)

C.0.2 Optimization with constrained length

To further study the effect of extraction threshold, we additionally apply summary size constraint on *summary output* heuristics while comparing them with *sentence output* heuristics with the according fixed thresholds. We show the results reported in average ROUGE scores[19] in Figure 2 with PEGASUS as the base generator. Apparently, *summary output* heuristics (i.e. *greedy* and *beam*) do not degenerate with excessive thresholds and typically discover better candidates compared to *sentence output* counterparts (*local* and *global*).

C.0.3 Criteria

To study the effect of different criteria, we re-execute *greedy search* with three evaluators: ROUGE-1, ROUGE-2 and sum of the two ROUGE-12[20]. We show the results in Table 25. The results are measured in an average of ROUGE scores[21]. In most scenarios, we observe that using ROUGE-1 leads to better results than related criteria.

C.0.4 Beam width

To explore the effect of different beam widths, we repeat the experiments with *beam search* while accounting for different beam values. The results are measured in an average of ROUGE scores[22] and presented in Table 26. Note that a beam size of 1 means the algorithm falls back to *greedy search*.

[19] ROUGE-1, ROUGE-2 and ROUGE-L

[20] These criteria are abbreviated as R-1, R-2 and R-12

[21] ROUGE-1, ROUGE-2 and ROUGE-L

[22] ROUGE-1, ROUGE-2 and ROUGE-L

Dataset	Model	R-1	R-2	R-12
CNN/DailyMail	PEGASUS	**32.88**	32.38	32.75
	BART	**33.26**	32.89	33.19
	BRIO	35.31	35.29	**35.47**
XSum	PEGASUS	**16.71**	15.59	16.68
	BART	**16.60**	15.53	16.54
	BRIO	**17.01**	15.82	17.00
RedditTIFU	PEGASUS	**16.65**	15.94	16.62
	BART	**17.71**	16.65	17.44
WikiHow	PEGASUS	**24.71**	23.33	24.54
	BART	**25.62**	24.36	25.56
PubMed	PEGASUS	**32.02**	29.62	31.31
	BART	**32.59**	30.11	31.94
	PRIMERA	**32.88**	30.75	32.38
Multi-News	PEGASUS	**35.16**	33.05	34.43
	BART	**35.12**	33.44	34.55
	PRIMERA	**36.73**	35.43	36.34

Table 25: Comparison between different criteria

Dataset	Model	1	4	8	16
CNN/DailyMail	PEGASUS	32.88	32.96	32.96	**32.97**
	BART	33.26	**33.39**	**33.39**	**33.39**
	BRIO	35.31	35.55	**35.58**	**35.58**
XSum	PEGASUS	16.71	**16.77**	**16.77**	**16.77**
	BART	16.60	**16.64**	**16.64**	**16.64**
	BRIO	17.01	**17.11**	**17.11**	**17.11**
RedditTIFU	PEGASUS	16.65	**16.69**	16.67	16.67
	BART	17.71	**17.75**	17.72	17.74
WikiHow	PEGASUS	24.71	24.74	**24.76**	24.74
	BART	25.62	25.63	**25.64**	**25.64**
PubMed	PEGASUS	**32.02**	31.97	31.97	31.97
	BART	**32.59**	32.55	32.56	32.56
	PRIMERA	32.88	**32.89**	**32.89**	32.88
Multi-News	PEGASUS	**35.16**	35.10	35.08	35.06
	BART	**35.12**	35.11	35.09	35.09
	PRIMERA	36.73	**36.75**	**36.75**	**36.75**

Table 26: Effect of different beam widths

For most cases, we observe that a beam size of 4 achieves good results and higher values do not significantly improve over it.

C.0.5 Inference with Large Language Model

Recent advances on large language models (LLMs) have unraveled emergent abilities (Schaeffer et al., 2023) that facilitate promising improvements in abstractive summarization (Zhang et al., 2024). To examine whether A2E can take advantage of these LLMs for extractive summarization, we reused and experimented with the corpus released by Zhang et al., 2024 which contains summaries generated by the *InstructGPT davinci v2* model (Ouyang et al., 2022) in zero- and few-shot ($k = 5$) in-context settings for 100 random samples in the CNN/DailyMall and XSum test sets. We present the details in Table 27 and 28. Under automatic evaluation, we find that A2E closely **approaches the abstractive summaries in CNN/DailyMall** and achieves **reasonable performance in XSum**. The summaries from A2E sometimes even achieve higher scores than the abstractive counterparts, e.g., the zero-shot results in CNN/DailyMall.

	R-1	**R-2**	**R-L**	**BERTScore**
Abstractive (Zero-shot)	37.05	13.72	34.42	62.03
Abstractive (Few-shot)	40.31	16.41	36.78	63.97
A2E Greedy (Zero-shot)	37.92	15.14	34.73	62.10
A2E Greedy (Few-shot)	39.61	16.81	35.78	62.98

Table 27: Zero-shot results with the *InstructGPT davinci v2* model on CNN/DailyMall.

	R-1	**R-2**	**R-L**	**BERTScore**
Abstractive (Zero-shot)	28.41	6.99	20.22	63.60
Abstractive (Few-shot)	34.87	12.97	26.37	67.84
A2E Greedy (Zero-shot)	21.04	3.50	16.40	56.65
A2E Greedy (Few-shot)	22.76	4.01	17.36	57.69

Table 28: Zero-shot results with the *InstructGPT davinci v2* model on Xsum.