Are Abstractive Summarization Models truly 'Abstractive'? An Empirical Study to Compare the two Forms of Summarization

Vinayshekhar Bannihatti Kumar
AWS AI Labs
vinayshk@amazon.com

Rashmi Gangadharaiah
AWS AI Labs
rgangad@amazon.com

Abstract

Automatic Text Summarization has seen a large paradigm shift from extractive methods to abstractive (or generation-based) methods in the last few years. This can be attributed to the availability of large autoregressive language models (Lewis et al., 2019; Zhang et al., 2019a) that have been shown to outperform extractive methods. In this work, we revisit extractive methods and study their performance against state of the art(SOTA) abstractive models. Through extensive studies, we notice that abstractive methods are not yet completely *abstractive* in their generated summaries. In addition to this finding, we propose an evaluation metric that could benefit the summarization research community to measure the degree of *abstractiveness* of a summary in comparison to their extractive counterparts. To confirm the generalizability of our findings, we conduct experiments on two summarization datasets using five powerful techniques in extractive and abstractive summarization and study their levels of abstraction.

1 Introduction

The amount of data on the internet has been growing exponentially creating excessive information for users to consume. Automatic text summarization alleviates this issue of information overload by producing shorter and concise summaries that capture the *essence* of the long source text. Automatic text summarization can be broadly classified into kinds- Extractive summarization and Abstractive summarization. Extractive summarization identifies important excerpts from the source document to produce summaries. These excerpts are composed of sentences and phrases which the model deems most appropriate for summarizing the source document. With the advent of sophisticated language models trained on large amounts of data, most of the recent work in summarization has drifted towards abstractive summarization. Summarization tasks have now become one of the benchmarks to beat with many of the SOTA generation models. In abstractive summarization, natural language generation techniques are employed to generate a summary.

Humans write concise summaries by introducing novel words and only using information from the source text that they deem absolutely necessary. Since abstractive summarization models generate words which are not necessarily present in the source text as is, the expectation is that the summaries generated would be truly *abstractive* and hence closer to human generated summaries. In theory, abstractive summarization models should outperform extractive methods as they have the ability to generate free form text like humans. In practice, however, we find that the abstractive summaries of SOTA models today are closer to extractive summaries than to the human generated summaries against which they are compared. Table 1 shows a generated summary by BART that has many overlapping snippets with the context. These observations have previously been unnoticed largely due to the evaluation methods that have been employed. As performing human evaluations is expensive, automatic metrics like BLEU (Papineni et al., 2002) and ROUGE (Papineni et al., 2002) have been used widely to measure the amount of overlap between generated summaries and reference summaries. As a result, these evaluation methods were not geared towards measuring *abstractiveness* of summaries. To better differentiate summaries generated by abstractive summarization models from their extractive counterparts, we also propose an evaluation metric called AbsExtScore.

In order to analyze the extent of these issues in abstractive summarization models, we chose three SOTA abstractive summarization models trained on CNN DailyMail (See et al., 2017) and the X-Sum (Narayan et al., 2018) datasets (Section 3). We also chose two SOTA extractive summarization

Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pages 198 - 206

Context	Abstractive Summary	Ground Truth
...prince george's first year - which boosted the economy by £247m...that a little girl could bring in £150m for the british economy...a princess would be able to set trends throughout her life...which will be great for the people who designed her clothes or those who can make quick knock-off copies	prince george's first year boosted the economy by £247m. a little girl could bring in £150m for the british economy. a princess would be able to set trends throughout her life. this will be great for the people who designed her clothes or those who can make quick knock-off copies.	as kate and william prepare to welcome their second child , there 's no doubt that the royal couple 's favoured childrenswear brands are hoping to cash in on the arrival .indeed , if prince george 's first year - which boosted the economy by 247m - is anything to go by , anything that the second-born touches will turn to gold and copycat designs will bring a welcome boost to the high street .the baby could generate a billion pounds over its lifetime .

Table 1: Example of Context, Ground truth and Abstractive Summary from CNN/DM dataset.

models trained on the same datasets (Section 4). We make the following observations and contributions:

- We observe that abstractive summarization models are not introducing enough novelty words and more simply copying over the words from the source text (Section 5).
- We observe that there is a large overlap between the extractive and abstractive summaries which questions the *abstractiveness* of the summaries generated by abstractive summarization models.
- To measure the quality of summaries in terms of *abstractiveness*, we propose an evaluation metric called AbsExtScore (Section 5.3). We also show that there is a correlation between human judgements and this evaluation metric by conducting a human subject study on a sample of summaries.

2 Related Work

Both extractive and abstractive summarization techniques have been well studied in the Natural Language Processing (NLP) community. Earlier work in extractive summarization relied on clues such as position of sentences and frequency of words while extracting most important snippets for summaries (Khan and Salim, 2014; Baxendale, 1958). More recently, neural network-based extractive summarization have gained more popularity (Alami et al., 2019; Xu and Durrett, 2019; Chen et al., 2018; Mohsen et al., 2020; Anand and Wagh, 2019; Zhong et al., 2020; Liu et al., 2019).

There has also been significant work in abstractive summarization (Genest and Lapalme, 2012; Barzilay et al., 1999; Tanaka et al., 2009). Most of the recent approaches use encoder-decoder architectures to generate summaries (Lee et al., 2020; Yao et al., 2020; Iwasaki et al., 2019; Zhang et al., 2019a; Raffel et al., 2019a; Lewis et al., 2019). These methods produce summaries using words that are not present in the source text and hence these methods have gained more popularity over the last few years. Transformer models with self-supervised training (Devlin et al., 2018; Radford et al., 2018; Raffel et al., 2019a; Yang et al., 2019; Clark et al., 2020; Liu et al., 2019) have shown to perform well on language learning when fine-tuned on various NLP tasks. More recently, BART(Lewis et al., 2019), PEGASUS (Zhang et al., 2019a) and T5 (Raffel et al., 2019b) have shown state-of-the-art performance in abstractive summarization, so we analyze the generated summaries of these models in comparison with SOTA methods in extractive summarization (Zhong et al., 2020; Liu et al., 2019). Previous work on measuring the abstractiveness and extractiveness in summaries has been restricted to measuring diversity of n-grams (Scialom et al., 2020; Grusky et al., 2018) but we move to a more semantic based metric to measure these aspects of a summary.

3 Datasets

We use two standard summarization benchmarks:
CNN-Dailymail (2017) : a corpus of news articles and human generated summaries. We use the entire test set of size $11,490$.
X-Sum (2018) : corpus of news articles and the task is to predict the first sentence given the remaining article content. We again use the entire test set of $11,334$ samples from this dataset.

4 Models

We study BART (Lewis et al., 2019), PEGASUS and T-5 (Raffel et al., 2019c) for abstractive summarization. For extractive summarization we use, BertSumExt (Liu, 2019) and MatchSum (Zhong et al., 2020). We use the versions fine-tuned on the datasets described in Section 3 for experiments. These models are in the top 10 on the leaderboard at the time of writing this paper [1]. These approaches are described below:
BART: An encoder decoder model that uses a bidirectional encoder and an autoregressive model as its decoder. The model is trained using de-noising objective.
PEGASUS: Encoder decoder model that is pre-trained to encourage abstractive summarization.

[1] https://paperswithcode.com/dataset/cnn-daily-mail-1

	CNN			XSum		
	Mean$_{[std]}$	**BLEU-1**	**BLEU-2**	**Mean**$_{[std]}$	**BLEU-1**	**BLEU-2**
ground truth	$0.751_{[0.098]}$	0.833	0.605	$0.519_{[0.158]}$	0.572	0.279
bart	$0.942_{[0.053]}$	0.955	0.903	$0.608_{[0.169]}$	0.661	0.383
pegasus	$0.889_{[0.063]}$	0.913	0.833	$0.584_{[0.174]}$	0.641	0.371
t5	$0.932_{[0.081]}$	0.937	0.849	$0.685_{[0.187]}$	0.726	0.446

Table 2: We measure the overlap between the source article and summaries by different models. Both the overlap metric defined in Section 5.1 and the BLEU metric without brevity penalty are shown in the tables. For the overlap metric we show both the mean(μ) and sigma(σ).

The model is trained to predict masked out sentences in the source text.

T-5: Text-based language problems are handled in a unified Text to Text framework to obtain significant improvements on several benchmark tasks.

BertSumExt: Model learns to pick the top 3 sentences that seem most relevant from the source text using Bert and inter-sentence transformer layers.

MatchSum: Model uses semantic matching between the document and the candidate summaries to pick the summary that is closest to the document. They achieve this using a margin-based triplet loss.

5 Experiments and Results

The summarization community has looked at using text overlap metrics like, BLEU or METEOR. The widely used text overlap metric for summarization is ROUGE-L that measures the overlap of words using Longest Common Subsequence between the generated summary and the ground truth. While these metrics attempt to compare the ground truth with the generated sentences, they certainly do not measure the *abstractive* component that one expects from abstractive summarization models. We describe experiments that show the model is merely copying words from the source text and not introducing novel words present in the ground truth. In order to allow future models to measure this abstractiveness we introduce a metric called *AbsExtScore* that will allow us to measure the abstractive prowess of these summarization models.

5.1 Source Text and Summary Overlap

We measure the percentage overlap of the words between the summary and the article for both the ground truth and the generated summaries, $overlap\_metric(J) = (S \cap T)/|T|$, S represents the set of source words and T represents the set of target words in the summary (be it ground truth or model generated). We also use the conventional BLEU metric to measure overlap between the source article and the target. However, we set the brevity penalty (BP) to 1 in order to reduce the penalty owing to the long sentences in the source article.

We see that the ground truth summaries have a much lower overlap with the article indicating that humans write a more abstractive version of the summary when compared to how a model performs abstractive summarization (Table 2). All the results are statistically significant using t-test. We see that there is at least 15% less overlap between the ground truth and generated summaries from any model on the CNN dataset and 8% less overlap on the XSum dataset. We also see that the overlap with BLEU is $10 - 15$ points higher for the generated summaries than the ground truth showing that the models are copying several words from the source text without actually introducing novel words.

5.2 Overlap between Abstractive and Extractive Summaries

We wanted to understand if the overlap between the abstractive summary and the ground truth was larger than the abstractive summary and the extractive summary. To investigate this, we measure the overlap between the summaries produced by 3 abstractive models and the summaries produced by 2 extractive models on 2 different datasets using Equation 1. However, in Table 3 we see that all the 3 abstractive models overlap more with the extractive summaries than the actual ground truth.

5.3 AbsExtScore

Abstractive summaries have to be closer to the ground truth in semantic space when compared to the extractive summary for any given source text. We use this as the foundational principle for our AbsExtScore which measures these distances in the semantic space. We project all the three summaries (abstractive, extractive and ground truth) into semantic space using the Siamese Distil Bert model (Reimers and Gurevych, 2019) trained on Bing queries. We use this model as it allows us to adapt well to different domains of source texts. Owing to the contrastive loss that this model was

	CNN			XSum		
	MatchSum	BertSumExt	Ground Truth	MatchSum	BertSumExt	Ground Truth
BART	$0.534_{[0.157]}$	$0.528_{[0.159]}$	$0.348_{[0.129]}$	$0.324_{[0.132]}$	$0.419_{[0.177]}$	$0.419_{[0.183]}$
PEGASUS	$0.547_{[0.157]}$	$0.532_{[0.16]}$	$0.381_{[0.142]}$	$0.309_{[0.134]}$	$0.411_{[0.18]}$	$0.447_{[0.194]}$
T5	$0.706_{[0.22]}$	$0.657_{[0.239]}$	$0.498_{[0.227]}$	$0.375_{[0.166]}$	$0.466_{[0.204]}$	$0.368_{[0.181]}$

Table 3: Overlap between the abstractive and extractive summaries. We see that the abstractive summaries overlap more with their extractive counterpart than with the ground truth.

	CNN						**XSum**					
	Eucledian Distance			**Cosine Similarity**			**Eucledian Distance**			**Cosine Similarity**		
	BART	**PEGASUS**	**T5**	**BART**	**PEGASUS**	**T5**	**BART**	**PEGASUS**	**T5**	**BART**	**PEGASUS**	**T5**
MatchSum	**0.72**	**0.663**	**0.615**	**0.724**	**0.668**	**0.629**	0.13	0.103	0.25	0.14	0.108	0.277
BertSumExt	**0.714**	**0.654**	**0.643**	**0.713**	**0.656**	**0.646**	**0.472**	0.405	**0.683**	**0.4661**	0.398	**0.674**

Table 4: AbsExtScore between different Abstractive and Extractive Summarization models. On CNN/DM we observe that both the Euclidean distance and cosine similarity favors the extractive summary over the ground truth for all scenarios. While on the XSum dataset it favors the ground truth in $3/6$ scenarios. We conducted a proportions z test and all results are statistically significant.

trained on, the model should be capable of capturing the difference in semantics of the three summaries. We define the score as below:

$$AbsExtScore = \frac{\sum_{n=1}^{N} argmin(d(e_{abs}, e_{gt}), d(e_{abs}, e_{ext}))}{N} \quad (1)$$

Here N is the total number of samples present in the test set. e_{abs}, e_{gt} and e_{ext} refers to embeddings obtained by using the MS Marco Distil Bert model (Reimers and Gurevych, 2019). d refers to the L2 distance between the two vectors. We also experimented with the use of a cosine similarity function. In this case we take $argmax$ as we want the vectors to be close to each other. We want the *AbsExtScore* to be close to 0 so that the abstractive vectors are closer to the ground truth vectors and not the extractive vectors. However, from Table 4 we see that in only $3/12$ different scenarios is the score less than 0.40, showing that the abstractive models overlap quite significantly with the extractive summaries in semantic space. We observe the same correlation with both the cosine similarity and Euclidean distance measures (L2 distance). Our metric is different from semantic measure metrics like BERTScore (Zhang et al., 2019b) and BARTScore (Yuan et al., 2021) as our metric captures both semantic similarity and 'abstractiveness' measures. We are not proposing a metric that measures the overlap between generated sentence and ground truth. We are only trying to introduce another dimensionality of measurement that helps answer the degree of abstractiveness of the model at a corpus level.

5.3.1 Human Subject study

To understand if there is a correlation between the automated metric proposed above and human judgement, we conducted a pilot human evaluation by randomly sampling 50 data points from the test set of CNN/DM. We presented the abstractive summary of this data point and asked the annotator to judge if it was closer to the ground truth over its extractive counterpart. We picked the BART abstractive model and MatchSum extractive model for this study. We got 3 annotations per datapoint (150 total). We said that *"An example of close resemblance includes but not limited to having similar phrases or having matching words."* to provide a judgement baseline to the annotators. We ran this study on Amazon Mechanical Turk. We removed a few annotations which were done in under 15 seconds (random annotations) . This reduced our total annotations to 62. Out of the 62 annotations, 41 said that the abstractive summaries are closer to extractive summary while 21 felt that it was closer to ground truth. We conducted a proportion z test to find that this result is statistically significant (p-value=0.007). Humans thought the abstractive summaries are close to extractive summaries 66% of the time while our metric gave a score of 72%. Agreement between humans and our metric was 76%.

6 Conclusion and Future Work

A central tenet of abstractive models is to abstract relevant information from source text. We find that the abstractive models merely copy words from source text and are failing to insert novelty words. We show that the abstractive summaries are closer to the extractive summaries than they are to ground truth. We use models built for semantic understanding to introduce a new metric called *AbsExtScore* which the summarization community can adopt to

understand the level of abstraction introduced by their proposed abstractive models in comparison to their extractive counterparts. We conducted a human subject study to show the correlation between the automated metric and human judgements.

Acknowledgement

We thank the anonymous reviewers for their valuable suggestions.

References

Nabil Alami, Mohammed Meknassi, and Noureddine Ennahnahi. 2019. Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. *Expert Syst. Appl.*, 123:195–211.

Deepa Anand and Rupali Sunil Wagh. 2019. Effective deep learning approaches for summarization of legal texts. *Journal of King Saud University - Computer and Information Sciences*.

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 550–557, College Park, Maryland, USA. Association for Computational Linguistics.

P. B. Baxendale. 1958. Machine-made index for technical literature—an experiment. *IBM Journal of Research and Development*, 2(4):354–361.

Kuan-Yu Chen, Shih-Hung Liu, Berlin Chen, and Hsin-Min Wang. 2018. An information distillation framework for extractive summarization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):161–170.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

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

Pierre-Etienne Genest and Guy Lapalme. 2012. Fully abstractive approach to guided summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 354–358, Jeju Island, Korea. Association for Computational Linguistics.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283*.

Yuuki Iwasaki, Akihiro Yamashita, Yoko Konno, and Katsushi Matsubayashi. 2019. Japanese abstractive text summarization using bert. In *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 1–5.

Atif Khan and Naomie Salim. 2014. A review on abstractive summarization methods. *Journal of theoretical and applied information technology*, 59:64–72.

Hyunsoo Lee, YunSeok Choi, and Jee-Hyong Lee. 2020. *Attention History-Based Attention for Abstractive Text Summarization*, page 1075–1081. Association for Computing Machinery, New York, NY, USA.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Farida Mohsen, Jiayang Wang, and Kamal Al-Sabahi. 2020. A hierarchical self-attentive neural extractive summarizer via reinforcement learning (hsasrl). *Applied Intelligence*, 50(9):2633–2646.

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. *arXiv preprint arXiv:1808.08745*.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019a. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019b. Exploring the limits of transfer learning with a unified text-to-text transformer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019c. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. Mlsum: The multilingual summarization corpus. *arXiv preprint arXiv:2004.14900*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Hideki Tanaka, Akinori Kinoshita, Takeshi Kobayakawa, Tadashi Kumano, and Naoto Kato. 2009. Syntax-driven sentence revision for broadcast news summarization. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation*, UCNLG+Sum '09, page 39–47, USA. Association for Computational Linguistics.

Jiacheng Xu and Greg Durrett. 2019. Neural extractive text summarization with syntactic compression.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.

Kaichun Yao, Libo Zhang, Dawei Du, Tiejian Luo, Lili Tao, and Yanjun Wu. 2020. Dual encoding for abstractive text summarization. *IEEE Transactions on Cybernetics*, 50(3):985–996.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

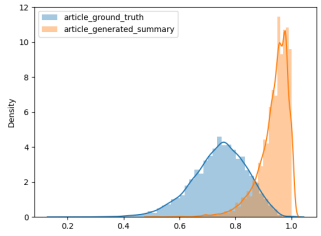
Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. *arXiv preprint arXiv:2004.08795*.

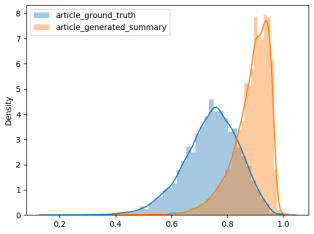
A Appendices

A.1 Overlap of Abstractive models with Source Text

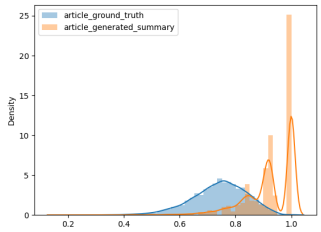
A.1.1 CNN/DM Dataset



(a) Bart

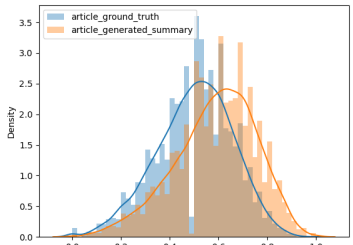


(b) Pegasus

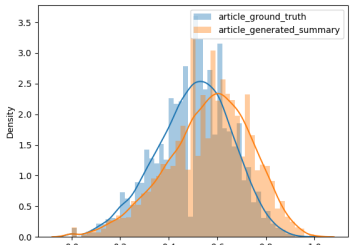


(c) T5

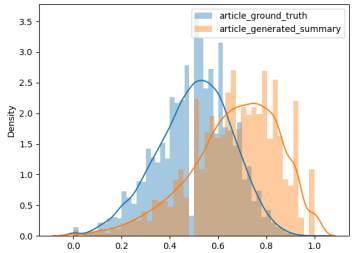
A.1.2 XSum Dataset



(a) Bart



(b) Pegasus

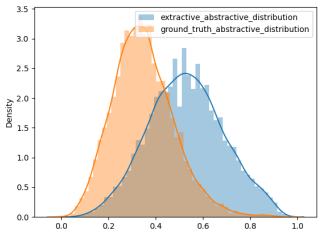


(c) T5

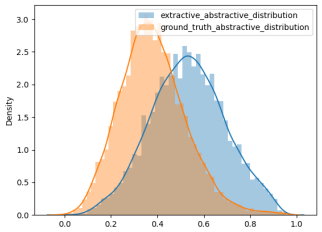
A.2 Overlap of Abstractive models with Extractive Models.

A.2.1 CNN/DM Dataset

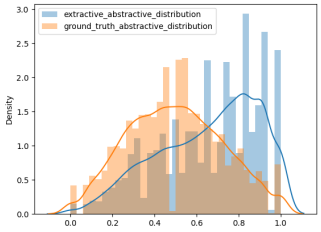
BertSumExt



(a) Bart

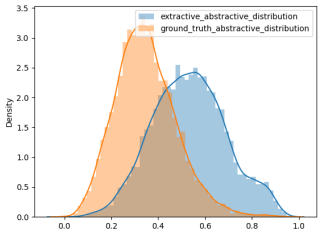


(b) Pegasus

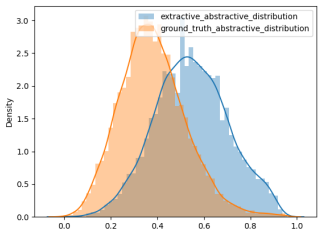


(c) T5

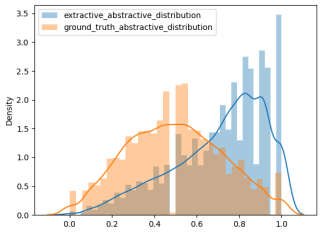
MatchSum



(a) Bart



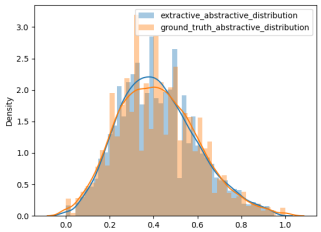
(b) Pegasus



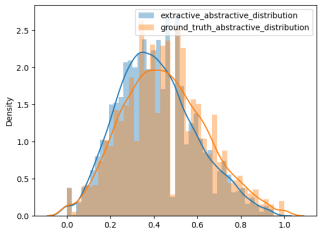
(c) T5

A.2.2 XSum Dataset

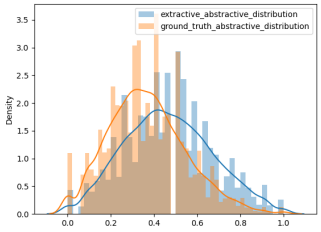
BertSumExt



(a) Bart

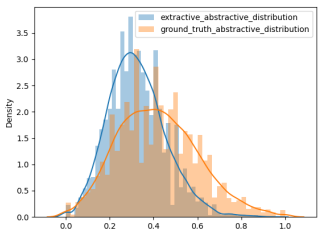


(b) Pegasus

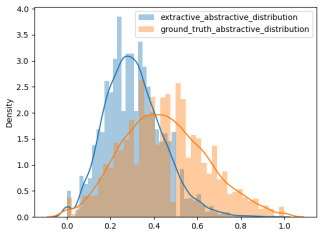


(c) T5

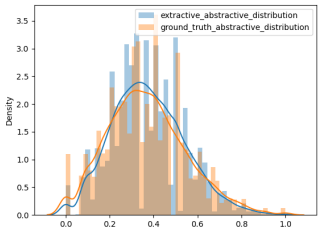
MatchSum



(a) Bart



(b) Pegasus



(c) T5