Indian Language Summarization using Pretrained Sequence-to-Sequence Models

Ashok Urlana, Sahil Manoj Bhatt, Nirmal Surange and Manish Shrivastava

Language Technologies Research Center, KCIS, IIIT Hyderabad, India

Abstract

The ILSUM shared task focuses on text summarization for two major Indian languages- Hindi and Gujarati, along with English. In this task, we experiment with various pretrained sequence-to-sequence models to find out the best model for each of the languages. We present a detailed overview of the models and our approaches in this paper. We secure the first rank across all three sub-tasks (English, Hindi and Gujarati). This paper also extensively analyzes the impact of k-fold cross-validation while experimenting with limited data size, and we also perform various experiments with a combination of the original and a filtered version of the data to determine the efficacy of the pretrained models.

Keywords

Indian language summarization, Sequence-to-Sequence models, Multilingual models,

1. Introduction

Automatic text summarization is a technique for obtaining a condensed version of a long document while retaining its relevance. The NLP community has become more interested in text summarization for Indian languages in recent years. The progress of text summarization has, however, been hindered due to the lack of high-quality datasets. Nevertheless, the availability of large-scale multilingual datasets such as XL-Sum[1] and MassiveSumm[2] have led to substantial progress in natural language generation and summarization tasks. Even though quality-wise, these datasets are far from perfect[3], they do serve as a good starting point in terms of quantity. Additionally, recent advancements in neural-based pretrained models have transformed the field significantly.

The goal of the ILSUM shared task is to create reusable corpora for Indian language summarization. The dataset is created by scraping the news articles and corresponding descriptions from publicly available news websites. ILSUM data[4, 5] consists of a summarization corpus for two major Indian languages- Hindi and Gujarati, along with Indian English.

This paper provides a comprehensive overview of the existing sequence-to-sequence models for Indian language and English summarization. For Hindi and Gujarati, we used multilingual models such as MT5[6], MBart[7] and IndicBART[8] variants. We fine-tuned the PEGASUS[9], BART[10], T5[11] and ProphetNet[12] models on English data. Out of all the models, for English, PEGASUS outperformed others, while for Hindi, MT5 gave us the best results, and for

Forum for Information Retrieval Evaluation, December 9-13, 2022, India

ashok.urlana@research.iiit.ac.in (A. Urlana); sahil.bhatt@research.iiit.ac.in (S. M. Bhatt); nirmal.surange@research.iiit.ac.in (N. Surange); m.shrivastava@iiit.ac.in (M. Shrivastava)

Gujarati, MBart performed the best. In order to avoid overfitting, we have performed k-fold cross-validation on the training dataset. We have observed that Hindi k-fold experiments had better scores than the experiments performed with the full version of the released data. We have applied several filters to assess the quality of the released datasets. Various combinations of filtered and original data were used to determine the efficacy of the pretrained generation models. We talk about our models, experiments and dataset filters later in this paper.

2. Related Work

Text summarization has been studied extensively, especially in the English language. Early research in summarization focused on extractive approaches, wherein summary sentences were chosen directly from the input text. On the other hand, abstractive approaches to summarization, such as neural attention models[13], Seq2Seq RNNs [14], Pointer-Generator networks [15] focus on generating summaries that capture the meaning of the input text without necessarily choosing sentences directly from the text. With the emergence of large neural language models for generation tasks, abstractive approaches have become more popular and generate high-quality summaries. While there have been various improvements in model architectures and summarization techniques, a large part of the progress in English text summarization can be attributed to the availability of large-scale datasets, such as CNN/DailyMail[14, 16], Gigaword[13, 17], XSum[18], etc.
This is in contrast to Indic languages, where little work has been done in summarization or related NLG tasks, such as headline generation. In recent times, however, there has been active research in this area, with the release of datasets such as XL-Sum[1], MassiveSumm[2], etc. These multilingual datasets consist of article-summary pairs from publicly available news domains, including Indian languages such as Hindi, Gujarati, Bengali, etc. The IndicNLG Suite[19] released datasets for several Indic language NLG tasks, such as sentence summarization and headline generation. More work needs to be done in this area to have models comparable to English summarization models in performance.

3. Corpus Description

The dataset released for this task has been collected from several leading Indian newspaper websites. The English and Hindi datasets were scraped from indiatvnews[1], and the Gujarati data was created by scraping the divyabhaskar[2] and gujarati.news18[3] websites. The Hindi and Gujarati datasets include articles/summaries which contain English words or phrases which have been code-mixed and script-mixed. Note that we have observed a few samples of English and Gujarati datasets, where the summaries consists of only one word. The ILSUM training data statistics are mentioned in Table 1. We have used the Indic[20] tokenizer to generate the counts in Table 1.

[1] https://www.indiatvnews.com/
[2] https://www.divyabhaskar.co.in/
[3] https://gujarati.news18.com/

Table 1
ILSUM Train Data Statistics

	English		Hindi		Gujarati	
#Pairs	12564		7957		8457	
	Text	Summary	Text	Summary	Text	Summary
#Avg Words	595	36.24	553	40.17	414.43	32.26
(Min, Max) Words	(1, 5717)	(1, 113)	(17, 5034)	(6, 113)	(25, 2839)	(1, 408)
#Avg Sentences	10.29	1.26	18.1	1.7	21.28	1.57
(Min, Max) Sentences	(1, 169)	(1, 17)	(1, 157)	(1, 9)	(1, 187)	(1, 46)

4. Model Description

The pretrained language models (PLMs) used for downstream tasks are pretrained using massive amounts of unlabeled text data. A PLM encodes extensive linguistic knowledge into a vast amount of parameters[21], which stimulates universal representations and improves generation quality. We have experimented with various pretrained generation models to find the optimal architecture.

T5 [11] model proposes defining every NLP task in a text-to-text format. The model consists of an encoder-decoder Transformer architecture finetuned on the C4 corpus. In our experiments, we use both the T5-Base (220M parameters) and T5-Large (770M parameters) versions of the model. Since T5 is trained on an English-only dataset, we also look at the multilingual variants of the model for our experiments in Hindi and Gujarati. The MT5 model[6] uses an architecture very similar to T5, and is trained on 101 languages, as described in the mC4 dataset. Owing to the large size of the models, we only finetuned the base version (580M parameters) of the MT5 model (the large version has 1.2B parameters).

BART [10] is a denoising autoencoder for pretraining seq2seq models, which is similar to both BERT and GPT. Since it uses a bidirectional encoder like BERT, and an autoregressive decoder like GPT. The model was trained by corrupting the text using a noising function, and reconstructing the original text. We experiment with the BART-large model (406M parameters), and then also try out versions of the BART model finetuned on different datasets, namely the BART-Large-CNN and BART-Large-XSUM model, finetuned on the CNN-Daily Mail and XSUM datasets respectively. We try out multilingual variants[7] of the BART model for Hindi and Gujarati summarization experiments, namely the MBart-Large-50 (610M parameters) model[22], trained on 50 languages.

PEGASUS [9] uses the extracted gap sentences (GSG) self-supervised objective strategy to train the encoder-decoder model. Rather than masking a smaller text span as in BART and T5, PEGASUS masks the entire sentence. Later, it concatenates the gap sentences into pseudo summaries. It chooses the sentences based on importance. In the same way as T5, PEGASUS does not reconstruct full sequence of inputs but only masked sentences. The pretraining is performed with C4[11] and HugeNews corpus. We finetune the PEGASUS-large model on the ILSUM English corpus.

BRIO [23] is a novel training paradigm to achieve neural abstractive summarization, wherein a contrastive learning component is introduced to reinforce the abstractive model's ability to estimate the probability of system-generated summaries more precisely instead of using MLE

training alone. Two stages are involved in this approach: the first stage generates the candidates using a pretrained sequence-to-sequence model, and next stage selects the best one.

ProphetNet [12] introduced a novel self-supervised objective, wherein the goal is to predict the next-n tokens, instead of just optimizing for one-step ahead predictions. We experiment with ProphetNet in our English summarization experiments.

IndicBART[8] is a pretrained sequence-to-sequence model trained on 11 Indic languages and English. It follows the masked span reconstruction objective similar to MBart. In contrast to available generation models, IndicBART utilizes the orthographic similarity between the Indian languages to achieve better cross-lingual transfer learning capabilities. This model size (244M) is much smaller than MBart and MT5 models with compact vocabulary. We finetune the IndicBART model on Hindi and Gujarati datasets.

Adapters: Recently proposed lightweight adapters[24] are effective at mitigating the overhead of pretrained language models for downstream tasks. We can update the adapters during finetuning and freezing most of the PLM parameters. In recent work[25], adapters were applied to perform Gujarati text summarization. Adapters can not only speed up training time but are also storage efficient since they require saving only adapter weights instead of entire finetuned model weights.

Table 2
ILSUM Experiments on Validation Data. *Finetuned on the combination of Hindi and Gujarati Data

			Validation Scores		
Lang	**Model**	**Full Data / k-fold**	**R-1**	**R-2**	**R-4**
English	PEGASUS	Full Data	**56.85**	**45.92**	**43.36**
	$T5_{large}$	Full Data	56.05	45.03	42.36
	$BART_{large}$	k-fold	54.83	43.58	40.71
	PEGASUS xsum	Full Data	54.66	43.48	40.64
	BRIO	Full Data	53.57	41.86	38.81
	$BART_{large}$ xsum	k-fold	53.35	41.74	38.75
	$T5_{base}$ + Adapter	k-fold	51.91	40.07	37.1
	ProphetNet	k-fold	49.51	36.98	33.83
Hindi	IndicBART	k-fold	**60.73**	**51.26**	**47.57**
	$MT5_{base}$	k-fold	60.04	50.72	46.82
	$MT5_{base}$*	Full Data	58.65	49.09	45.08
	IndicBART-SentSumm	k-fold	58.09	47.99	43.72
	$MBart_{large}$50 + Adapters	Full Data	56.26	45.56	41.21
	$MBart_{large}$50	Full Data	55.76	44.96	40.59
Gujarati	$MBart_{large}$50	Full Data	**26.20**	**16.44**	**12.16**
	$MT5_{base}$	Full Data	25.11	15.81	11.68
	$MT5_{base}$*	Full Data	24.16	14.68	10.79
	IndicBART	k-fold	23.38	13.34	9.35
	$MBart_{large}$50 + Adapter	Full Data	21.63	13.04	9.56

5. Experiments and Results

We have performed experiments under two different settings: the first is with the entire released dataset (full data), and the other is where we split the dataset into 10 folds and utilize 90% data (9 folds) for training and 10% data (1 fold) for validation. In both settings, the released data in the validation phase was used for testing purposes and we report these results in Table 2. Note that doing such k-fold cross validation experiments were also essential to evaluate our models' performance because validation summaries were not provided to us.
We use the standard ROUGE metric[26] to compute all the scores. We observed that PEGASUS yields the best results for English when finetuned on the full data version in the validation phase. We achieved the best results when we finetuned IndicBART and MBart using k-fold and full data during the validation phase. Finetuning a model on k-fold data might sometimes lead to better results than finetuning it on the entire dataset, which indicates that the dataset needs to be studied more and appropriate filters need to be applied, to see which examples in the dataset contribute to the model learning something useful. We discuss this in the next section. Based on the results of the validation phase, we submit results from the best models in the test phase. While PEGASUS and MBart still give us the best results for English and Gujarati respectively, MT5 performs better than IndicBART for Hindi when finetuned on k-fold data. Hyper-parameter settings are listed in Table 4.
The multilingual models have been pretrained on large amounts of data, and they are sufficiently capable of handling the presence of code-mixing in the dataset, which we observe in the outputs as well. The models generate good summaries and can add relevant English text in Hindi and Gujarati examples where appropriate. For instance, the average number of English words in Hindi and Gujarati training summaries is 0.25 and 1.91 respectively. For the test set released for Hindi and Gujarati, the summaries generated by our models have an average of 0.23 and 1.44 English words per summary. Note that the average number of English words in Hindi summaries is less because a large number of training samples are purely in Hindi and do not contain any English words or characters.

Table 3
ILSUM scores on Test Data

			Test Scores		
Lang	**Model**	**Full Data / k-fold**	**R-1**	**R-2**	**R-4**
English	PEGASUS	Full Data	**55.83**	**44.58**	**41.8**
	$T5_{large}$	Full Data	54.73	43.08	40.12
Hindi	$MT5_{base}$	k-fold	**60.72**	**51.02**	**47.11**
	IndicBART	k-fold	58.38	48.31	44.25
Gujarati	$MBart_{large}50$	Full Data	**26.11**	16.51	12.41
	$MBart_{large}50$	Full Data (dropout=0.2)	26.07	**16.60**	**12.58**

6. Data Quality Assessment

To verify the quality of the data, we have applied some of the filters mentioned in TeSum[3]. Filters were applied include checking whether there are:

Table 4
Experimental setup and parameters settings

Parameters	BART	T5	ProphetNet	PEGASUS	BRIO	MBart	MT5	IndicBART
Max source length	512	512	512	512	512	512	512	512
Max target length	75	75	75	75	75	75	100	75
Batch Size	2	1	1	2	2	4	2	2
Epochs	5	5	5	5	5	5	10	10
Vocab Size	50265	32128	30522	96103	50264	250054	250112	64015
Beam Size	4	4	5	4	4	4	4	4
Learning Rate	5e-5	5e-5	5e-5	5e-4	5e-5	5e-5	5e-5	5e-5

1. Empty instances
2. Duplicate pairs and summaries within the dataset
3. Cases where the first few sentences of the article itself are taken as the summary
4. Check whether the summary is 'compressed enough', i.e., we should not have summaries comparable in size to the text that has to be summarized. Compression is a good measure of telling us if the summary provided is a shortened version of the input document/text or not.

Filters counts for all the languages can be found in Table 5. It is important to note that, based on our filters, only about 68% of the Hindi summaries are valid since many are simply the first few sentences of the article. It could also be one of the reasons for models giving better results on k-fold data. Some of the folds in the training data might contain a large percentage of high-quality, valid summaries while leaving out a significant number of summaries which we consider invalid. Note that for Gujarati and English, the number of final valid article-summary pairs is comparable to the original dataset size, which is why the top-performing models give better results when finetuned on the whole dataset as compared to k-fold subsets.

Table 5
Filtration counts of ILSUM data

Filters	Hindi	Gujarati	English
Dataset Size	7957	8457	12565
Empty	0	0	1
Duplicate Pairs	23	0	0
Duplicate Summary	15	113	117
Prefixes	2518	135	486
Compression <50%	11	37	182
Final Valid	5390	8172	11779
Valid %	**67.74%**	**96.63%**	**93.74%**

6.1. Data Variation Experiments

The unavailability of large datasets is one of the main bottlenecks for neural models for text generation. The existing summarization datasets for Indian languages are quite small. To improve the model generation capabilities on limited dataset, we did k-fold cross-validation

Table 6
Validation set ROUGE scores on ILSUM corpus. This table reports the mean ROUGE scores and its standard deviation over 10 runs

Lang	Model	Data composition	R-1	R-2	R-L
English	PEGASUS	Original Data	52.51 ± 1.1	40.91 ± 1.36	47.81 ± 1.16
		Original + Filtered Data	51.65 ± 1.14	40.07 ± 1.25	46 ± 3.67
		Filtered Data	51.88 ± 1.25	40.37 ± 1.39	47.32 ± 1.31
		Filtered + Original Data	53.28 ± 1.18	41.82 ± 1.3	48.67 ± 1.2
	T5-large	Original Data	**53.45 ± 0.95**	**42.16 ± 1.13**	**48.97 ± 1.05**
		Original + Filtered Data	53.22 ± 1.23	42.04 ± 1.41	48.85 ± 1.31
		Filtered Data	51.9 ± 1.37	40.49 ± 1.53	47.38 ± 1.46
		Filtered + Original Data	53.33 ± 0.83	42.1 ± 0.96	48.92 ± 0.86
	BART-large	Original Data	50.25 ± 1.52	38.15 ± 1.85	45.46 ± 1.63
		Original + Filtered Data	51.42 ± 0.88	39.85 ± 1.11	46.93 ± 1
		Filtered Data	51.21 ± 1.3	39.83 ± 1.57	46.79 ± 1.38
		Filtered + Original Data	52.45 ± 1.05	40.98 ± 1.29	48 ± 1.17
Hindi	IndicBART	Original Data	26.36 ± 1.02	12.66 ± 0.73	26.28 ± 0.98
		Original + Filtered Data	21.58 ± 0.66	9.84 ± 0.76	21.45 ± 0.6
		Filtered Data	21.27 ± 0.88	9.75 ± 0.56	21.12 ± 0.86
		Filtered + Original Data	25.67 ± 1.04	12.16 ± 0.82	25.57 ± 1
	MT5-base	Original Data	**27.04 ± 1.22**	**13.21 ± 0.61**	**26.96 ± 1.22**
		Original + Filtered Data	20.33 ± 0.91	9.26 ± 0.8	20.2 ± 0.92
		Filtered Data	20.61 ± 1.55	9.47 ± 0.67	20.51 ± 1.53
		Filtered + Original Data	26.73 ± 1.11	12.83 ± 0.61	26.64 ± 1.1
Gujarati	MBart Large 50	Original Data	20.36 ± 0.67	11.65 ± 1.13	20.01 ± 0.72
		Original + Filtered Data	16.04 ± 1.12	9.23 ± 0.76	15.83 ± 1.15
		Filtered Data	12.82 ± 2.28	6.6 ± 1.54	12.38 ± 2.36
		Filtered + Original Data	19.55 ± 0.74	11.42 ± 0.43	19.2 ± 0.72
	MT5-base	Original Data	**21.55 ± 0.77**	**11.81 ± 0.78**	**21.19 ± 0.83**
		Original + Filtered Data	18.63 ± 0.93	9.23 ± 0.5	18.19 ± 0.92
		Filtered Data	9.66 ± 0.97	4.84 ± 0.56	9.53 ± 0.92
		Filtered + Original Data	20.29 ± 0.62	10.7 ± 0.52	19.84 ± 0.56

on the best performing models (see Table 2). The mean ROUGE scores and standard deviation scores over 10 runs are reported in Table 6. We did 10-fold cross-validation using the released training dataset with the following combinations:

1. **Original data:** Fine-tuned for 5 epochs with released training dataset
2. **Original + Filtered data:** Finetuned for 3 epochs with original + 2 epochs with Filtered data
3. **Filtered data:** Fine-tuned for 5 epochs with only filtered dataset
4. **Filtered + Original data:** Finetuned for 3 epochs with filtered data + 2 epochs with original data

To perform all the experiments, we used the 'filtered data' obtained after applying filters mentioned in Table 5. To compare the models' performance on different variations of the training dataset, we have not made any changes in the validation data. As observed in Table 6,

the experiments performed with 'original' data produce better scores than the 'filtered' data. Also, the models finetuned on the combination of the 'filtered + original' dataset performed better compared to the 'original+filtered' combination.

7. Discussion and Conclusions

While having better models finetuned exclusively on Indian languages might benefit research in the area of Indian Language Summarization, creating larger, high-quality datasets for such languages will surely lead to progress in this field. It might be interesting to look at sources other than news websites as well, and to keep in mind the filters discussed earlier while creating the dataset.
For the ILSUM task, PEGASUS, MT5 and MBart give us the best results for English, Hindi and Gujarati respectively. We conclude that the transformer-based pretrained seq2seq models are capable of generating high-quality summaries for the ILSUM shared task.

Acknowledgements

We thank the organizers of the ILSUM shared task for their help and support.

References

[1] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Samin, Y.-F. Li, Y.-B. Kang, M. S. Rahman, R. Shahriyar, Xl-sum: Large-scale multilingual abstractive summarization for 44 languages, arXiv preprint arXiv:2106.13822 (2021).

[2] D. Varab, N. Schluter, MassiveSumm: a very large-scale, very multilingual, news summarisation dataset, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 10150–10161. URL: https://aclanthology.org/2021.emnlp-main.797. doi:10.18653/v1/2021.emnlp-main.797.

[3] A. Urlana, N. Surange, P. Baswani, P. Ravva, M. Shrivastava, Tesum: Human-generated abstractive summarization corpus for telugu, in: Proceedings of the Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 5712–5722. URL: https://aclanthology.org/2022.lrec-1.614.

[4] S. Satapara, B. Modha, S. Modha, P. Mehta, Fire 2022 ilsum track: Indian language summarization, in: Proceedings of the 14th Forum for Information Retrieval Evaluation, ACM, 2022.

[5] S. Satapara, B. Modha, S. Modha, P. Mehta, Findings of the first shared task on indian language summarization (ilsum): Approaches, challenges and the path ahead, in: Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, Kolkata, India, December 9-13, 2022, CEUR Workshop Proceedings, CEUR-WS.org, 2022.

[6] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mt5: A massively multilingual pre-trained text-to-text transformer, arXiv preprint arXiv:2010.11934 (2020).

[7] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, L. Zettlemoyer, Multilingual denoising pre-training for neural machine translation, Transactions of the Association for Computational Linguistics 8 (2020) 726–742.

[8] R. Dabre, H. Shrotriya, A. Kunchukuttan, R. Puduppully, M. M. Khapra, P. Kumar, Indicbart: A pre-trained model for natural language generation of indic languages, arXiv preprint arXiv:2109.02903 (2021).

[9] J. Zhang, Y. Zhao, M. Saleh, P. Liu, Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, in: International Conference on Machine Learning, PMLR, 2020, pp. 11328–11339.

[10] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, arXiv preprint arXiv:1910.13461 (2019).

[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al., Exploring the limits of transfer learning with a unified text-to-text transformer., J. Mach. Learn. Res. 21 (2020) 1–67.

[12] W. Qi, Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen, R. Zhang, M. Zhou, Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training, arXiv preprint arXiv:2001.04063 (2020).

[13] A. M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, CoRR abs/1509.00685 (2015). URL: http://arxiv.org/abs/1509.00685. arXiv:1509.00685.

[14] R. Nallapati, B. Xiang, B. Zhou, Sequence-to-sequence rnns for text summarization, CoRR abs/1602.06023 (2016). URL: http://arxiv.org/abs/1602.06023. arXiv:1602.06023.

[15] A. See, P. J. Liu, C. D. Manning, Get to the point: Summarization with pointer-generator networks, CoRR abs/1704.04368 (2017). URL: http://arxiv.org/abs/1704.04368. arXiv:1704.04368.

[16] K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, CoRR abs/1506.03340 (2015). URL: http://arxiv.org/abs/1506.03340. arXiv:1506.03340.

[17] D. Graff, J. Kong, K. Chen, K. Maeda, English gigaword, Linguistic Data Consortium, Philadelphia 4 (2003) 34.

[18] S. Narayan, S. B. Cohen, M. Lapata, Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization, ArXiv abs/1808.08745 (2018).

[19] A. Kumar, H. Shrotriya, P. Sahu, R. Dabre, R. Puduppully, A. Kunchukuttan, A. Mishra, M. M. Khapra, P. Kumar, Indicnlg suite: Multilingual datasets for diverse nlg tasks in indic languages, 2022. arXiv:2203.05437.

[20] A. Kunchukuttan, The IndicNLP Library, https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf, 2020.

[21] J. Li, T. Tang, W. X. Zhao, J.-R. Wen, Pretrained language models for text generation: A survey, arXiv preprint arXiv:2105.10311 (2021).

[22] Y. Tang, C. Tran, X. Li, P. Chen, N. Goyal, V. Chaudhary, J. Gu, A. Fan, Multilingual translation with extensible multilingual pretraining and finetuning, CoRR abs/2008.00401 (2020). URL: https://arxiv.org/abs/2008.00401. arXiv:2008.00401.

[23] Y. Liu, P. Liu, D. Radev, G. Neubig, Brio: Bringing order to abstractive summarization, arXiv preprint arXiv:2203.16804 (2022).
[24] J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, I. Gurevych, Adapterhub: A framework for adapting transformers, arXiv preprint arXiv:2007.07779 (2020).
[25] Z. Zhao, P. Chen, To adapt or to fine-tune: A case study on abstractive summarization, arXiv preprint arXiv:2208.14559 (2022).
[26] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.