Comparative analysis of T5 model for abstractive text summarization on different datasets

Tawmo, Mrinmoi Borah, Pankaj Dadure, Partha Pakray

National Institute of Technology Silchar
Assam, India 788010

Abstract

Abstractive Text Summarization is a burgeoning natural language processing task that has seen success with the Transformer model. With the soaring amassment of unstructured text data, text summarization has found applications in many domains such as law, medicine and news. The growth in research interest in developing better summarization systems, more fluent human-like summaries, and significant, more refined datasets have offered to test natural language processing models to improve existing model performance. This paper investigates the T5 Transformer model for abstractive text summarization and analyses its performance on the CNNDM, MSMO and XSUM datasets. The proposed model compared the resultant output across the datasets to determine the proficiency of the model and the datasets with regards to ROUGE and BLEU scores.

Keywords: Text Summarization, Abstractive Summary, News articles, T5, Transformer

1. Introduction

Text information is ever-increasing online, from social media posts, tweets, blogs, newsletters and research papers being some of the agents contributing to the accumulating text data. Finding meaningful information from the sea of data in a limited period can be difficult. A solution to this problem can be Automatic Text Summarization(ATS) which facilitates the generation of short paraphrased articles of the source document. Automatic text summary generation can aid in increasing productivity and reducing time consumption with the shortened texts articles produced without manual intervention. Fields such as law, medicine, news, and blogs where a large quantity of written information is frequently generated can benefit from Automatic Text Summarization applications. Based on the type of summarization approach, ATS can be classified into extractive and abstractive text summarization. Extractive text summarization [1][2][3][4] method summarises the source text by copying or extracting the important sentence from the source verbatim. In contrast, in the Abstractive [5][6][7] manner, the source text is paraphrased and compressed to generate a summary with novel sentences which might not have been in the source article, much like how a human would write a summary. The focus of research has been on the extractive approach of summarization which is faster and simpler with higher accuracy than the abstractive approach [8]. But due to the direct extraction of phrases from the source document, the summaries generated can have redundant [9], incoherent [10] and lengthy [11] sentences. In comparison, the abstractive approach of summarization generates superior summaries, concise and comprehensive, but it is comparatively more difficult [9] as natural language generation technology is a necessary component.

Interest in researching abstractive text summarization has spawned various models and methods. There are three classifications of abstractive text summarization methods 1) structure-based that employs predefined structures such as trees and graphs, 2) Semantic-based that uses semantic representations of texts, 3) based on deep learning techniques [12][13][14][15][16][17] which has been the centre of focus of research. Sequence to sequence deep learning model made abstractive summary practical and achievable. Transformers [18] was initially introduced as a sequence to

Email addresses: ttawmo@gmail.com (Tawmo), mrinmoiborah2@gmail.com (Mrinmoi Borah), krdadure@gmail.com (Pankaj Dadure), parthapakray@gmail.com (Partha Pakray)

sequence machine learning model and later adopted in many other applications of NLP, such as summarization and question answering. The success of transformer-based pre-trained models in the NLP task has stimulated various models such as T5 [19], GPT [20] and BERT [21].

The Text-to-Text Transformer (T5) was introduced in [19], where the input are strings and the model outputs modified strings. The main contribution of this paper is to investigate the performance of the T5 model for the task of abstractive text summarization on different datasets. To perform comparative analysis on the proficiency T5 model, CNNDM, MSMO and XSUM datasets have been used, and experimental results have been shown in ROUGE and BLEU scores. The manuscript discusses the related works in section 2. It describes the dataset used in the proposed model in section 3. Section 4 is about the system architecture, and section 5 describes the experimental setup. Section 6 discusses the results analysis and conclusion and future research direction is reviewed in section 7.

2. Related Work

Automatic text summarization was researched first by Luhn [1] fifty years ago using the statistical technique; sentences consisting of highest frequency and lowest frequency words were not considered essential and less likely to be included in the summary. Edmunson [3][2] also explored includes scoring a sentence's importance by considering a linear combination of features such as the position, word frequency, cue words and the structure of the document, sentences which has the highest weight were to be included in the summary. Abstractive text summarization is more complex than the extractive method because the abstraction-based summarization task involves language generation to form a coherent synopsis. Deep-learning based abstractive summarization was first implemented by Rush et al. [13] in 2015 based on the encoder-decoder architecture. The RNN Encoder-Decoder architecture was proposed by Cho et al. [22], which can learn from the mapping sequence of a variable length to another sequence of a variable length. Similarly, Sutskever et al. [23] used multilayered Long Short-term Memory (LSTM) to map the source sequence to a fixed dimensionality vector and then from the vector decode the target sequence with another deep LSTM. The model proposed by [22] and [23] was further extended on by Bahdanau et al. [12]. They pointed out that compressing all the information of the source sequence into a fixed-length vector may lead to the neural network having difficulty coping as the source sequence length increases. They addressed this potential issue by letting a model search for a set of input words when generating each target word. Abstractive summarization produces more fluent summaries but requires more computational power than extractive summarization and is more complex. In recent years, transformer architecture [24][21][25] has led to significant advancements in various NLP tasks. Zhang et al. [26] showed that Pegasus outperformed the state of the art models. Pegasus creates a pseudo-summary by selecting and masking entire sentences from texts, then concatenating the gap sentences. To accomplish abstractive summarising, pre-training leverages extracted gap sentences and sequence-to-sequence. ProphetNet [27] is based on the Transformer [18] encoder-decoder architecture which demonstrated state-of-the-art results for abstractive summary experimented on CNNDM and Gigaword corpus. Another transformer based model by Zaheer et al. Mastropaolo et al.[28] used the pretrained T5 model for automatic code summarization and bug fixes. More recently, Du et al. [29] proposed GLM (General Language Model) to address the challenge of models not performing well on all main categories of natural language understanding, conditional generation, and unconditional generation at the same time also achieving model produced State-of-the-art results.

To evaluate the quality of the system produced summary, various evaluation methods are employed, such as ROUGE [30], BLEU [31] and many more [32]. It is vital to have a reliable metric for the summary to evaluate the algorithms for automatic summarization.

3. Dataset

The CNN/Daily-mail (CNNDM) and MSMO datasets have been used to test, train and validate the proposed model. The CNNDM dataset has been collected from the website of Huggingface[1]. The CNNDM dataset contains 2,86,817 news articles that have been used for training the model. For frequent evaluation, 13,368 documents have been used as validation data, and finally, 11,490 documents have been used for testing. Each document of the dataset

[1] https://huggingface.co/datasets/cnn_dailymail

Table 1: Dataset Description

Data	CNNDM	MSMO	XSUM
Train	2,86,817	2,93,625	204,045
Validation	13,368	10,339	11,332
Test	11,490	10,295	11,334

contains approximately 28 sentences pair. In addition to this, we have used the MSMO dataset, which has been collected from the website of NLPR[2]. The MSMO dataset contains news articles that have a body, title, and summaries in it. We have used the body as our source data and the title as summary data. The MSMO dataset contains 2,93,625 article bodies and titles as a source and target data that has been used for training. For testing, 10,295 articles have been used, and for validation, we have used 10,339 articles. In addition, we experimented with the XSUM dataset presented in [33], collected from a repository [3].The Extreme Summarizing(XSUM) [33] dataset is a collection of abstractive single-document summarization algorithms that can be evaluated to develop a new one-sentence synopsis. 226,711 new articles were compiled from BBC articles published between 2010 and 2017 and cover various topics (e.g., News, Politics, Sports, Weather, Business, Technology and Entertainment etc). In training, validation, and test sets, the official random split contains 204,045 (90%), 11,332 (5%), and 11,334 (5%), respectively. The dataset statistics have been presented in Table 1.

4. System Architecture

In the field of natural language processing, transfer learning [34] has become more prevalent in recent years. To pre-train models on large datasets, strings tokens masking and generating the masked token is the foundation in a self-supervised task. The trained model is then fine-tuned on smaller, more specialised datasets, each intended to serve a specific activity. The T5 model was proposed by Raffel et al., which achieves the state of the art performance on various NLP tasks; the model was trained on a large-scale dataset that was finetuned for numerous downstream tasks.

The T5 model is based on the transformer architecture, which was proposed by Vaswani et al. [18]. The architecture utilised the stack of self-attention layers instead of conventional RNNs and CNNs, enabling the support of variable-sized inputs. The transformer model is composed of encoder-decoder layers connected to a multi-head attention layer followed by a feed-forward network, as illustrated in Figure 1. Layer normalisation is performed to each subcomponent's input. In the completes stacks input and output, the feed-forward network, the skip connection and in the attention weights, dropout is implemented. Similar to how the encoders function, in decoders also have self-attention layers as well as structures which pays attention to the encoder's output. To construct the output probabilities over the vocabulary, the output of the last decoder block is forwarded to a dense layer additionally with a softmax output. The difference between generic transformer models and the T5 transformer is that the T5 model uses a reduced position embedding. Each embedding is a scalar added to the relevant logit used to compute the attention weights. T5 transformer has encoders that are similar in structure. The model was first trained in Transfer Learning on a task with a massive corpus before being fine-tuned on a downstream task to learn general-purpose tasks like summarization, machine translation, and caption generation. From the five proposed T5 versions: T5-small, T5-base, T5-large, T5-3b and T5-llb, in this paper, we have used the T5-base version.

[2]http://www.nlpr.ia.ac.cn/cip/dataset.htm
[3]https://github.com/EdinburghNLP/XSum

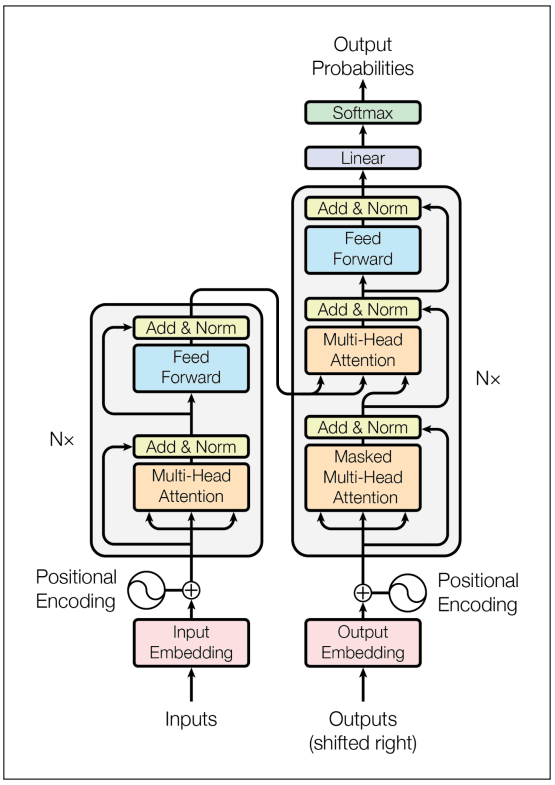

Figure 1: Transformer Model Architecture [18]

5. Experiment

5.1. Preprocessing

The datasets mentioned in section 3 have been cleaned before feeding the model. The datasets have been converted to lower case, any special characters or punctions such as "# $ % & ˜ _ { } . , = @ \ / [ ] - + _ : ; ' * ! ?" were removed. We have also expanded short forms of words such as "he'll, she'll, I'll, it'd" etc. to "he will, she will, I will, it would" etc. which is called contraction mapping.

5.2. Hyperparameters

We have experimented with the T5-base version of the model. The model has been trained and tested on the CNNDM, MSMO and XSUM datasets described in section 3. The pre-trained model has been fine-tuned, and the results of the three datasets are compared. The model has been implemented with the Pytorch Lighting and Transformer Library. We define the input target length of 512 tokens and the output summary length of 128 tokens. Adam optimizer with a learning rate of 1e-3 is used to optimize the model. The model has been trained for a total of 1 epoch with a batch size of 8. The proposed model has used three different datasets, and for each dataset, the model has deployed and tuned an equal number of hyperparameters. It has compared the resultant output across the datasets to determine the proficiency of the model and the datasets.

6. Result Analysis

It is vital to have a reliable metric for the summary to evaluate the algorithms for automatic summarization. We have employed the Bilingual Evaluation Understudy (BLEU) [31] and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [30] metric to assess the system-generated summary's calibre. BLEU is a score that was initially

proposed for comparing a source and a reference translation which is now also implemented to evaluate machine-generated summaries, while ROUGE score is a defacto metric for assessing the automatically generated summaries. Table 2 & 3 shows the ROUGE score and BLEU score of each dataset respectively. Figure 2, 3 & 4 shows the document level BLEU score analysis of the CNNDM, MSMO and XSUM dataset respectively. Figure 5, 6 & 7 illustrates the document level ROUGE score analysis of CNNDM, MSMO and XSUM datasets.

- The MSMO dataset has accomplished higher ROUGE and BLEU scores of 42.287 and 43.9, respectively, compared to the summaries generated from the CNNDM and MSMO datasets.
- The sentence-level BLEU score on the three datasets by our proposed model shows a similar distribution, i.e., the majority of summaries have scored between 25 to 50.
- Similary, the ROUGE score distribution on sentence-level follows an identical pattern with majority of the sentence score between 25 and 50.
- 29.7% of the summaries have a BLEU score above 50 with the MSMO dataset, whereas the CNNDM and the XSUM dataset have 25.81% and 14.65% of the summaries that achieved a BLEU score more than 50.

Table 2: Obtained ROUGE score

Data	CNNDM	MSMO	XSUM
ROUGE-1	40.791	42.287	35.063
ROUGE-2	18.551	20.411	12.597
ROUGE-L	34.80	34.579	27.527

Table 3: Obtained BLEU score

Data	CNNDM	MSMO	XSUM
BLEU-1	43.5	43.9	38.3
BLEU-2	19.4	21.3	11.8
BLEU-3	11.6	11.7	5.4
BLEU-4	8.0	7.3	2.7
BLEU	14.58	14.18	7.31

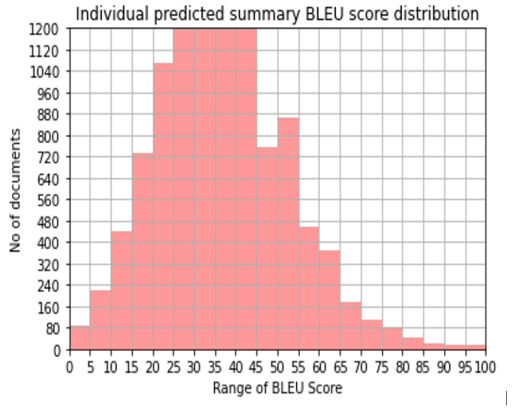

Figure 2: BLEU score analysis on document level of CNNDM dataset

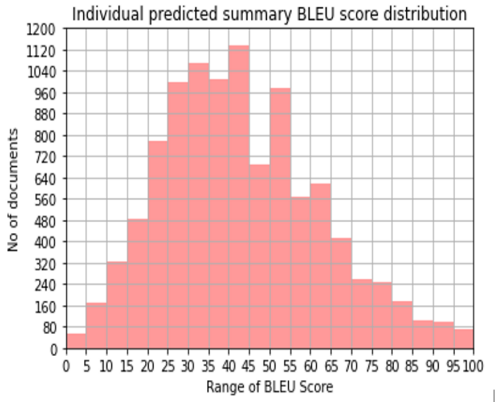

Figure 3: BLEU score analysis on document level of MSMO dataset

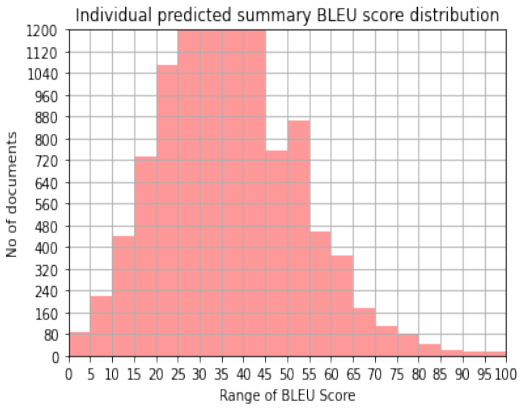

Figure 4: BLEU score analysis on document level of XSUM dataset

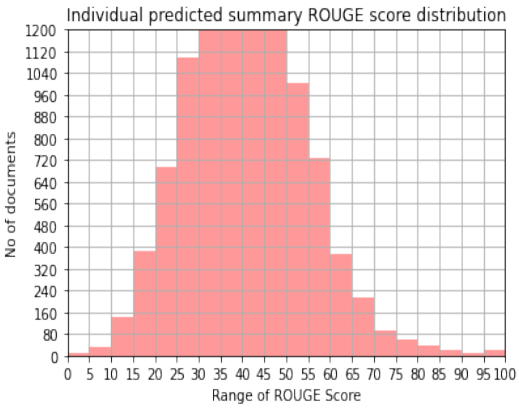

Figure 5: ROUGE score analysis on document level of CNNDM dataset

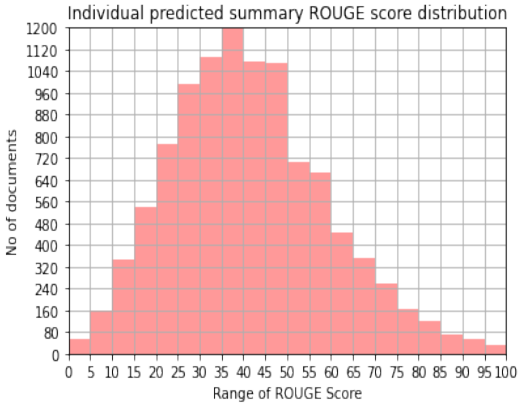

Figure 6: ROUGE score analysis on document level of MSMO dataset

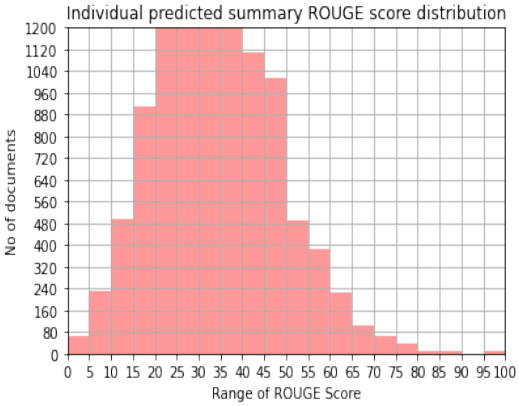

Figure 7: ROUGE score analysis on document level of XSUM dataset

7. Conclusion

In this paper, we have analysed the performance of the T5 model on three datasets: CNNDM, MSMO and XSUM for abstractive summarization. Our findings showed that the pretrained T5 transformers produced excellent short abstracts, resulting in a sound and fluent summary for particular text material. For comparative analysis, we calculated ROUGE and BLEU scores for proposed the model's performance on each dataset's predictions and determined that the T5 model accomplished the best result with the MSMO dataset. Future work includes exploring more robust models that can generate more concise, fluent and coherent abstractive summaries and comparing the performance of various transformer-based models on the explored datasets.

Acknowledgement

The work presented here falls under the Research Project Grant No. IFC/4130/DST-CNRS/2018-19/IT25 (DST-CNRS targeted program). The authors would like to express gratitude to the Centre for Natural Language Processing and Artificial Intelligence Lab, Department of Computer Science and Engineering, National Institute of Technology Silchar, India for providing infrastructural facilities and support.

References

[1] H. P. Luhn, The automatic creation of literature abstracts, IBM Journal of research and development 2 (1958) 159–165.

[2] H. P. Edmundson, R. E. Wyllys, Automatic abstracting and indexing—survey and recommendations, Communications of the ACM 4 (1961) 226–234.

[3] H. P. Edmundson, New methods in automatic extracting, Journal of the ACM (JACM) 16 (1969) 264–285.

[4] D. R. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, et al., Mead-a platform for multidocument multilingual text summarization (2004).

[5] I. F. Moawad, M. Aref, Semantic graph reduction approach for abstractive text summarization, in: 2012 Seventh International Conference on Computer Engineering & Systems (ICCES), IEEE, pp. 132–138.

[6] S. Song, H. Huang, T. Ruan, Abstractive text summarization using lstm-cnn based deep learning, Multimedia Tools and Applications 78 (2019) 857–875.

[7] H. Saggion, T. Poibeau, Automatic text summarization: Past, present and future, in: Multi-source, multilingual information extraction and summarization, Springer, 2013, pp. 3–21.

[8] A. Tandel, B. Modi, P. Gupta, S. Wagle, S. Khedkar, Multi-document text summarization - a survey, in: 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), pp. 331–334.

[9] L. Hou, P. Hu, C. Bei, Abstractive document summarization via neural model with joint attention, in: National CCF Conference on Natural Language Processing and Chinese Computing, Springer, pp. 329–338.

[10] N. Moratanch, S. Chitrakala, A survey on extractive text summarization, in: 2017 international conference on computer, communication and signal processing (ICCCSP), IEEE, pp. 1–6.

[11] V. Gupta, G. S. Lehal, A survey of text summarization extractive techniques, Journal of emerging technologies in web intelligence 2 (2010) 258–268.

[12] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).

[13] A. M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, arXiv preprint arXiv:1509.00685 (2015).

[14] S. Chopra, M. Auli, A. M. Rush, Abstractive sentence summarization with attentive recurrent neural networks, in: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 93–98.

[15] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al., Abstractive text summarization using sequence-to-sequence rnns and beyond, arXiv preprint arXiv:1602.06023 (2016).

[16] A. See, P. J. Liu, C. D. Manning, Get to the point: Summarization with pointer-generator networks, arXiv preprint arXiv:1704.04368 (2017).

[17] Q. Chen, X.-D. Zhu, Z.-H. Ling, S. Wei, H. Jiang, Distraction-based neural networks for modeling document., in: IJCAI, volume 16, pp. 2754–2760.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[19] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, arXiv preprint arXiv:1910.10683 (2019).

[20] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).

[21] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[22] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[23] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, pp. 3104–3112.

[24] 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).

[25] 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).

[26] 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, pp. 11328–11339.

[27] 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).

[28] A. Mastropaolo, S. Scalabrino, N. Cooper, D. Nader Palacio, D. Poshyvanyk, R. Oliveto, G. Bavota, Studying the usage of text-to-text transfer transformer to support code-related tasks, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), pp. 336–347.

[29] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, J. Tang, All nlp tasks are generation tasks: A general pretraining framework, arXiv preprint arXiv:2103.10360 (2021).

[30] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, pp. 74–81.

[31] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311–318.

[32] S. Banerjee, A. Lavie, Meteor: An automatic metric for mt evaluation with improved correlation with human judgments, in: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp. 65–72.

[33] S. Narayan, S. B. Cohen, M. Lapata, 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.

[34] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (2009) 1345–1359.