Recent Developments in Automatic Explanation of Information Visualizations

Anonymous EMNLP submission

Abstract

Information Visualizations such as bar graphs, pie chart, time series graphs are an easy and effective way for communicating insights. In scientific community, visualizations are sometimes complex and include several variables. This can be challenging for people who have difficulty with understanding visualizations. To tackle this, there has been some research done on automatic generation of textual descriptions of charts. This comes under the process called Natural Language Generation (NLG). In this article, I aim to explain 1) Natural Language Generation from non-linguistic input i.e. Data; 2) Compare End-to-End vs Pipelined architectures for Data-to-Text NLG; 3) Give an overview of three Chart-to-Text datasets proposed in recent years; 4) Look at some of the recently proposed NLG evaluation metric. At the end, I mention some of the future directions we can take for developing better Chart-to-Text models by leveraging large pre-trained transformer models.

1 Introduction

Natural Language Generation is an important research problem for many NLP applications. The recent advances in Natural Language Generation were due to large Pre-trained text generation Transformers (Vaswani et al., 2017) like GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2019) and BART (Lewis et al., 2019). Without the exception of GPT-2, all three models are Sequence to Sequence (Sutskever et al., 2014), meaning that they expect input as a sequence or text, and give output in text. It is still debated whether for a textual output generation, the input should be in a linguistic format (Gatt and Krahmer, 2018).

Considering a Sequence-to-Sequence architecture, the above mentioned models were adapted and fine-tuned on the task of Converting Data to Text. The input to the model can be in various forms like database records, spreadsheets, JSON file and so on. In this Data-to-Text problem, we look at Chart-to-Text; Generating descriptions or summaries for charts and see what advances have been made.

Chart to Text has many applications in academia and industry. It can be used by researchers to generated analytical descriptions of their results for publication. For the visually impaired, it can be used as a medium to effectively communicate information visualization.

2 Data to Text

The goal of Data-to-text is to generate text given some structured data. The problem of NLG w.r.t Data-to-Text is divided into several subproblems (Reiter and Dale, 1997).

1. **Content Selection**: Decide what information will be used to generate text.
2. **Text Planning:** How this information should be structured
3. **Sentence Planning:** Decide how the information will be aggregated in a sentence and what words and phrases will be used.
4. **Realization:** Generate well formed sentences.

Pipelined approaches perform all the above mentioned steps and on some abstract level end-to-end approaches do that as well. Figure 1 shows a pipelined system proposed by Reiter and Dale (1997). Each module can be rule-based or statistical including neural. Figure 2 shows an End-to-End system proposed by Gehrmann et al. (2018).The End-to-End architecture is a Sequence-to-Sequence model. The Encoder (on the left) reads the data and generates context representations. These context representations are fed parallelly into two decoders (on the right). The decoders generate the text in parallel and the one that generates the better text gets a parameter update.

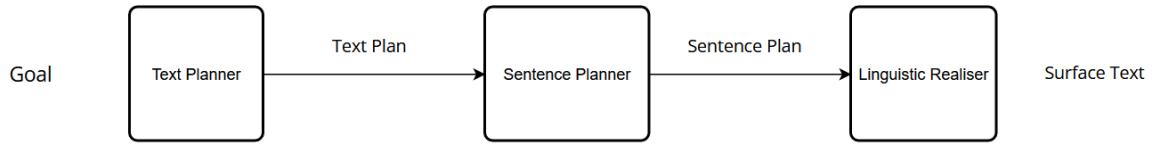


Figure 1: A Pipelined NLG System

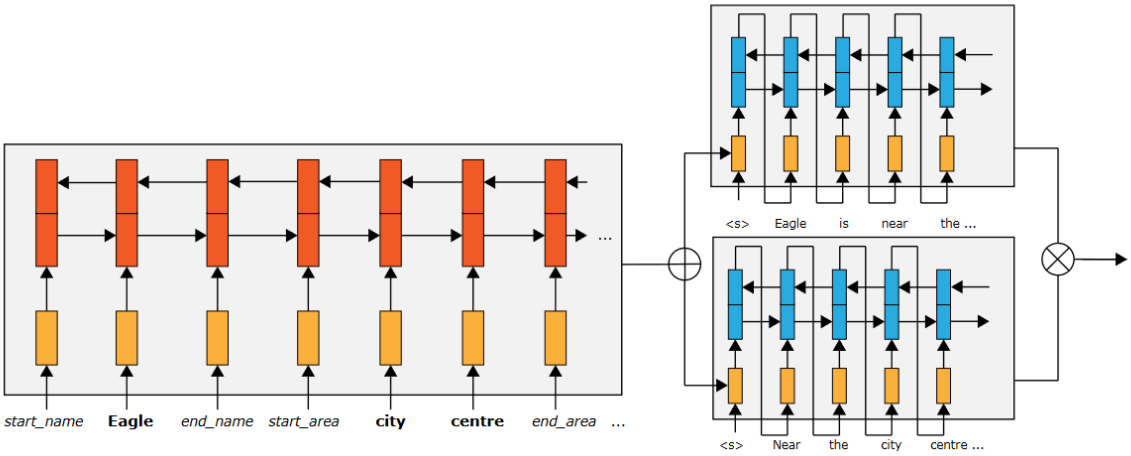


Figure 2: An End-to-End NLG System

2.1 Pipelined

Based on the above subproblems, Ferreira et al. (2019) proposed a 5 sequential step modular approach. It consists of **Discourse Ordering**, **Text Structuring Lexicalization** or Sentence Planning, **Referring Expression Generation** which involves generating the references to the entities of discourse (Krahmer and Van Deemter, 2012), and **Textual Realization**. Each component was tested on two Neural architectures: Gated Recurrent Unit - GRU (Cho et al., 2014) Neural Network with attention mechanism (Vaswani et al., 2017) and an encoder-decoder transformer.

Ferreira et al. (2019) found during the analysis of pipelined components that GRU works better at Ordering and structuring of non-linguistic input while transformer is better at verbalizing the structured set of triples. GRUs, unlike transformer which take the input all at once, read the text sequentially which could help them perform better at discourse ordering and planning step.

2.2 End-to-End

Encoder-Decoder (Sutskever et al., 2014) architectures use End-to-End training. Gehrmann et al. (2018) proposed an encoder-decoder model that takes linearized RDF triples and converts them to text.

2.3 End-to-End vs Pipelined

Ferreira et al. (2019) found that pipelined approach is better than End-to-End as it produces high quality text for all domains, especially the un-seen ones. End-to-End approach is not good at generalizing and it often suffers with hallucinations (Rohrbach et al., 2018); a phenomena where the generated text contains representations that are not present in the input. So adding supervision to sub-components of Data-to-Text generation will generated fluent, coherent text without hallucinations.

Another important thing is controllablity. Controllability allows the end user to control certain aspects of the text at the time of generation like style, length, sentiment etc. Controllability is easier to enforce in pipelined approach.

We can theorize from the findings that End-to-End approach did not perform well because neither, GRUs and transformers, are better than each other in all five components (2.1). For example, It is possible that when an End-to-End transformer model

Chart Type	Line	Bar	Total
Simple	3564	3199	6763
Complex	902	640	1542
Total	4466	3839	8305

Table 1: Chart-to-Text: Distribution

Chart Type	Line	Bar	Scatter
Temporal	1929	3085	1929
Categorical	951	1387	951
Total	2880	4472	2880

Table 2: AutoChart: Distribution

decides to do the initial ordering and text structuring (at some abstract level), it struggles and it carries an ill-formed representation to the realization level.

We can also hypothesize based on the findings of Wang and Chen (2020) on Positional Embeddings. Transformer models include positional embeddings to encode order of the sequence. The directionality of encoding of the information (Unidirectional or Bidirectional) is based on the training objective [1]. Wang and Chen (2020) found that transformers based on Masked Language Models (Devlin et al., 2018) learn local position while the autoregressive models learn absolute position[2]. What this suggests us with regards to End-to-End vs Pipelined debate, is that End-to-End models could perform better if they were trained on autoregressive training objective. Autoregressive training objective is theoratically much closer to the way GRU is trained (in a sequential manner).

3 Chart-to-Text

For Chart-to-Text to be modelled as Data-to-Text problem, some data needs to be extracted from the Charts that can be fed into the model as structured data. Such data usually includes X-Y labels, title and type of the chart/graph, height or trend of a variable, legends and so on. To generate data for charts/graphs, opensource tools like ChartReader (Rane et al., 2021), PDFFigures 2.0 (Clark and Divvala, 2016), are used. ChartReader classifies the chart, detects and extracts text, X-Y label, legend, and data. The tool consists of machine learning, and rule based algorithms that perform all the above mentioned functions. PDFFigures 2.0, takes an entire document as an input and extracts figures, tables, and captions. This tool utilizes text classification heuristics and unsupervised clustering for detecting figure regions. This tool can be used to further extract any text inside a figure.

Over the past two years, three large parallel Chart-Summary pair datasets have been created and made available to the public. Obeid and Hoque (2020) created a dataset crawled from statista.com, Zhu et al. (2021) created a template based dataset of Charts and analytical summaries, and Hsu et al. (2021) created a dataset of scientific figures collected from arXiv along with captions.

4 Datasets and Approaches to the problem

4.1 Char2Text (Obeid and Hoque, 2020)

Obeid and Hoque (2020) introduced a dataset on chart summarization where the summaries were written by humans. Table 1 shows the distribution of figures. A simple line and bar chart contains just one line and or a set of bars. A complex line and bar chart contains more than one lines in a graph and stacked bar charts. Figure 3 shows one of the chart-summary pair along with the data that is fed as an input.

Obeid and Hoque (2020) adapted a Data-to-Text transformer proposed by Gong et al. (2019). This model extended the standard transformer by adding a binary prediction layer and a content selection training step. This model expects a tuple of 4 features (entity, type, value, information) and then generates the output based on the latent representation generated by the encoder.

Chart-to-Text authors proposed several changes to the input of Data-to-Text transformer so it can be adapted for Chart Summarization. First, they modified the four feature records to (column header, table cell value, column index, chart type). Secondly, they reintroduced the positional embeddings because information visualization often contain ordered relationships. Lastly, data variable substitutions are used to tackle the issue of hallucinations. The transformer trained on Chart2text data resulted with a BLEU score of 18.54.

4.2 AutoChart

One of the issues with generating Chart-Summary dataset is that the Summaries in the dataset are

[1] For example, BERT's training objective is Masked Language Modeling and GPT-2's training objective is autoregressive language modeling.

[2] **Absolute position** encodes the absolute position of a unit within a sentence. **Relative Position** encodes the position of a unit relative to other units.

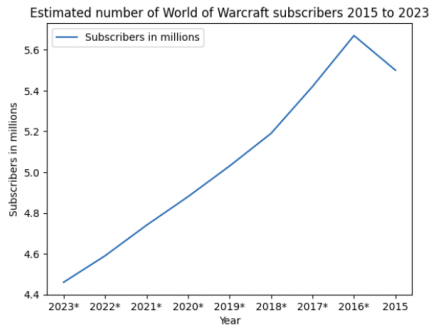
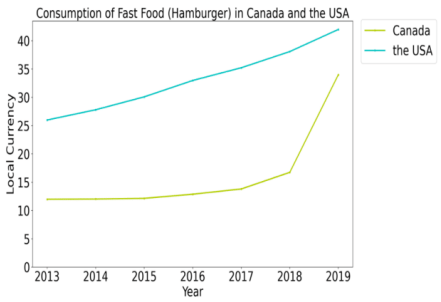
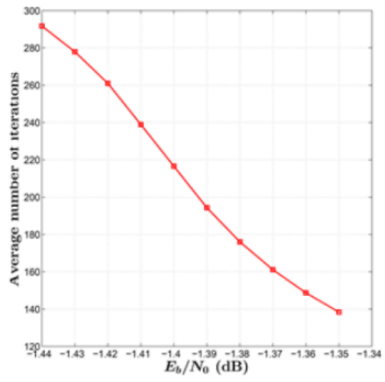
Dataset	Image	Summary/Description
Chart2Text		**Summary:** How many people play World of Warcraft ? In 2015 , when Activision Blizzard last reported on WoW 's subscriptions , the game had 5.5 million global subscribers . It is projected that the numbers will be gradually decreasing to reach 4.46 million in 2023 . The game reached the peak of its popularity in the second half of 2010 , when Activision Blizzard 's estimates put the global number of WoW subscribers at 12 million **Data:** Year|2023|x|line_chart Subscribers_in_millions|4.46|y|line_chart Year|2022|x|line_chart Subscribers_in_millions|4.59|y|line_chart Year|2021|x|line_chart Subscribers_in_millions|4.74|y|line_chart Year|2020|x|line_chart Subscribers_in_millions|4.88|y|line_chart Year|2019|x|line_chart Subscribers_in_millions|5.03|y|line_chart Year|2018|x|line_chart Subscribers_in_millions|5.19|y|line_chart Year|2017|x|line_chart Subscribers_in_millions|5.42|y|line_chart Year|2016|x|line_chart Subscribers_in_millions|5.67|y|line_chart Year|2015|x|line_chart Subscribers_in_millions|5.5|y|line_chart
AutoChart		The line graph displays the number of consumption of fast food (hamburger) in Canada and the USA, respectively, from 2013 through 2019. In this chart, the unit of measurement is Local Currency, as seen on the y-axis. The data related to Canada is rendered yellow and the cyan line is for the USA. It is obvious that both countries shared similar increasing trends in the number of consumption in the past 6 years.For Canada, by 2013 the number of consumption reached nearly 12, while the number continued to increase until 34 in 2019.And for the USA, in 2013, the number of consumption was about 26, after that, each year has witnessed some increase. In 2019, the number of consumption was about 42. In the past 6 years, the USA had consistently more than Canada. With 13 for Canada and 35 for the USA in 2017, a new record of large difference was registered.It would be interesting to see what would happen in the next decade in these two countries in terms of current situations.
SciCap		Average number of iterations required to decode PLDPC-Hadamard code with r=8 and k= 204, 802.

Figure 3: Chart-Summary pairs.

Data Collection	Model	BLEU-4
First Sentence	CNN + LSTM	2.05
	LSTM	2.13
Single Sentence Caption	CNN + LSTM	2.02
	LSTM	2.12
Caption with <= 100 words	CNN + LSTM	1.68
	LSTM	1.65

Table 3: Experimental results of model trained on the three dataset collections.

not analytical. The summaries quote figures from the charts/graphs but do not analyze the chart and provide insights to it. Another issue is based on Data-to-Text models. There is an additional step that researchers have to do in order to automate Summary generation of chart/graphs. That step is extracting data from Charts and structuring it.

Zhu et al. (2021) tries to address both the issues by proposing a template based Chart-Summary creation. Using their framework; AutoChart, they generated analytical summaries of three kinds of charts; Line, Bar (Vertical and Horizontal), and Scatter plots. Their proposed framework contains two modules; **chart generation** and **analytical description generation**. Figure 4 shows the complete flow of Chart-Summary pair generation process. First, statistical data, which includes different variables whose relation could be plotted, is collected. Then a trend strategy is formulated where data perturbation is applied to generate various types of trends. Statistical Data and Trend strategy is then fed into the two separate modules to generate Chart-Description pairs.

Table 2 shows distribution of figures. 10,232 figures have been generated with 23,543 summaries. Meaning that there are multiple summaries for each figure. The reason the authors did this is to simulate real world where there can be multiple acceptable analytical summaries of each chart figure.

You can look at figure 3 and see how the summary of the AutoChart example is much more detailed and informative as compared to Chart2Text, and contains only chart relevant information. In Chart2Text example, the last sentence of the summary contains information that is not present in the chart.

4.3 SciCap

Hsu et al. (2021) also tried to address the issue of Data Driven summary generation and generic descriptions of Charts/graphs by constructing SciCap. This dataset was constructed using computer science papers from arXiv. It is a large dataset containing 2 million figures from over 200,000 papers. SciCap includes three different types of Data collections. Each sampled using a different strategy.

1. **First Sentence**: Collection includes figures with caption of only the first sentence.

2. **Single Sentence Caption**: Single sentence complete captions.

3. **Caption with No more than 100 words**: Complete captions of length less than hundred words. Average length of caption is 1.66 sentence.

Figure 3 shows an example from SciCap dataset. The provided summary is rather small. It is more of a caption than a summary. The caption only tells what the graph is about. It does not describe the X-Y labels and analyzes it. There are many examples in this dataset with smaller descriptions and descriptions of low quality. The reason behind could be that most of the figures and their corresponding description or rather, captions, in the the computer science papers only tell what the graph is about and rest of the description of the graph is mentioned in the actual text of the paper. The authors of SciCap also provided us with baselines where they used Data-to-Text and Vision-to-Text approaches. In Data-to-Text, they used PDFFigure 2.0 (Clark and Divvala, 2016) to extract data from the figures and trained an Encoder-Decoder model to generate captions. During the preprocessing of training a Data-to-Text model, they normalized or delexicalized the captions by removing chart specific information. For Vision-to-Text, they used CNN+LSTM (Xu et al., 2015) based encoder-decoder. Image and text vector is concatenated and fed into the LSTM Decoder to generate the outputs. Their results concluded that using Data-to-Text yielded slightly

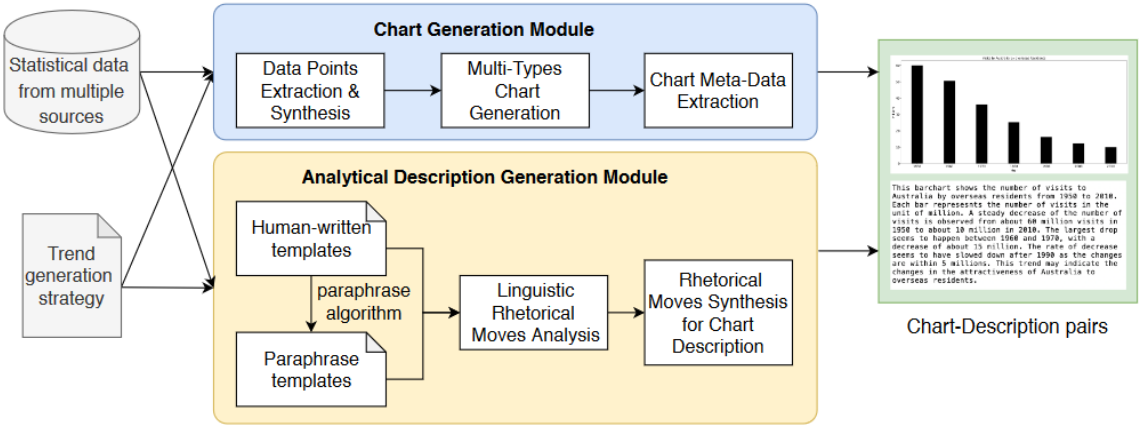


Figure 4: AutoChart Generation Process.

better results compared to Vision-to-Text.

Table 3 shows the comparison of models trained on SciCap dataset on different data collections.

5 Problems of Alignment

Parallel Corpora needs to be aligned. Alignment is the task of finding correspondence between a source and a target text (Legrand et al., 2016). For example, in a machine translation task, words, sentences or entire documents are aligned between source and target text. Similarly, for Data-to-Text task, the elements mentioned in the data needs to be aligned with their references in target summary. Let us say that the target summary does not mention the correct title of the chart (from the data), we will say that the chart-summary pair is misaligned.

Issue of misalignment occur during the creation of the dataset. This can be automatically resolved by string matching or can be done manually (Iza Skrjanec, 2022). Misalignments can also occur when processing real-time time series graphs and real-time comments need to be generated. Hamazono et al. (2020) addresses this issue by introducing a multi-timestep encoder-decoder model with copy-mechanism. They improve upon the work of Hamazono et al. (2020) and modified the generic encoder so it takes multiple input vectors and equipped the decoder with copy mechanism which facilitates learning correspondences between data and text from noisy training data.

6 Evaluation Metrics

NLG is evaluated in two ways; automatically and by humans. Automatic evaluations help us evaluate our models quickly and thus help us in hyper parameter tuning during the training phase. Human evaluation is important because the candidate text will never be exactly similar to the reference text. So to evaluate the quality and faithfulness of the text to the input, human evaluation is important.

6.1 Automatic Metrics

On a high level, generating chart summaries is a Sequence-to-Sequence task like Machine Translation or Automatic Summarization. Because of this, the most common evaluation metric used here is BLEU - BiLingual Evaluation Understudy (Papineni et al., 2002). This method measures the n-gram overlap of reference text and generated/hypothesis/candidate text. A sister metric of BLEU is called ROUGE - Recall Oriented Understudy of Gisting Evaluation (Lin, 2004). It is most commonly used for Automatic Summarization but can be used for Chart Summarization as well as it only measures n-gram recall.

The problem with BLEU and ROUGE is that they rely on surface level similarity and exact matches. To overcome these drawbacks, BERTscore (Zhang et al., 2019) was proposed which is a learned evaluation metric based on pre-trained BERT contextual embeddings. Contextual embeddings are effective when capturing distant dependencies and ordering. Even though BERTScore is a significant improvement from ROUGE and BLEU, it is not error-free. Hanna and Bojar (2021)

showed that BERTscore fails to assign score when a bad candidate has high lexical overlap with reference in terms of content words [3]. Another learned evaluation metric for text generation is BLEURT (Sellam et al., 2020) which evaluates quality of a text. In simple terms, it is basically BERT for evaluation. BLEURT was evaluated on the WebNLG challenge and it outperformed BLEU, ROUGE and BERTScore, and achieved high correlation with human ratings.

6.2 Manual Evaluation

Manual evaluations are another way of evaluating the quality of the text. For this, humans go through the text and provide their feedbacks in the form of ratings. The rating can be based on a simple question like '*Which of the two summaries best explain the above chart?*' or they can be fine-grained and ask the annotator to rate on the basis of several factors like coherence, fluency, consistency, title relevance, etc.

Another important test is input fidelity. It is important that the generated summary describes the chart data and provides insights according to the data. Input fidelity can be low if the model generates hallucinations.

7 Future Directions

The area of Neural Chart Summary generation is new. There are many challenges that need to be addressed. First challenge is evaluation of datasets on a single neural Data-to-Text architecture and see which dataset trains the best model. Secondly, the power of pre-trained language models like GPT-2 and T5 is not utilized so far. Pre-trained language models are trained in a self supervised manner on large dataset so that makes them *well read*. They are trained on a large corpus like wikipedia so they can figure out the patterns in the language themselves. Later, when they are fine-tuned on downstream tasks like text summarization, they generate fluent, coherent text. Conversely, models trained only on downstream task like the ones trained by the authors of chart-to-text and SciCap, need to understand language and the task at the same time.

GPT-2 and T5 can be fine-tuned on Chart Summaries. Clive et al. (2021) uses T5 and achieves SOTA on several Data-to-Text datasets inlcuding WebNLG. Peng et al. (2020) uses a modified version of GPT-2 called SC-GPT which is designed for few shot learning of dialog acts. Kale and Rastogi (2020) shows that linearized version of Data can be given as input to T5. Pre-trained on several Data-to-Text datasets, T5 leads to SOTA performance and greatly improves results on out-of-domain inputs.

Lastly, what needs to be looked at is the Vision-to-Text task. Instead of modelling the problem as data-to-text, we can model the problem as vision-to-text, or an image captioning task. This has further challenges related to how the visual hidden representations can be translated to textual representations. A Visual Transformer (Dosovitskiy et al., 2020) was proposed where the image was fed into the transformer encoder as small patches (similar to tokens in NLP) in a sequential manner. The hidden representations generated by ViT were used as an input to GPT-2(with cross-attention module) as showed by Luo et al. (2022). Visual Conditioned GPT (VC-GPT) achieves either the best or the second best results on image captioning datasets.

8 Conclusion

In this paper, I first gave a brief overview of Data-to-Text and compared Pipelined vs End-to-End architectures. After that, I talked about the latest chart summarization datasets and the approaches that various researchers have used to address the problem. Then, I talked about NLG Evaluation metrics. Lastly, I make a case for utilizing pre-trained transformers in the future for Chart Summarization. The area of Chart Summary generation is relatively new and under-explored. Large datasets have been made public only in the past one year. Hopefully, researchers will use and develop better architectures for this task with the availability of these datasets. I also look forward to creation of a real world interactive application that gives a textual description of a chart.

[3] Content Words: Adjective, Adverbs, Noun, Main Verb

References

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Christopher Clark and Santosh Divvala. 2016. Pdffigures 2.0: Mining figures from research papers. In *2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL)*, pages 143–152.

Jordan Clive, Kris Cao, and Marek Rei. 2021. Control prefixes for text generation. *arXiv preprint arXiv:2110.08329*.

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*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Thiago Castro Ferreira, Chris van der Lee, Emiel Van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. *arXiv preprint arXiv:1908.09022*.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Sebastian Gehrmann, Falcon Z Dai, Henry Elder, and Alexander M Rush. 2018. End-to-end content and plan selection for data-to-text generation. *arXiv preprint arXiv:1810.04700*.

Li Gong, Josep Crego, and Jean Senellart. 2019. Enhanced transformer model for data-to-text generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 148–156, Hong Kong. Association for Computational Linguistics.

Yumi Hamazono, Yui Uehara, Hiroshi Noji, Yusuke Miyao, Hiroya Takamura, and Ichiro Kobayashi. 2020. Market comment generation from data with noisy alignments. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 148–157, Dublin, Ireland. Association for Computational Linguistics.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of BERTScore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, Online. Association for Computational Linguistics.

Ting-Yao Hsu, C Lee Giles, and Ting-Hao'Kenneth' Huang. 2021. Scicap: Generating captions for scientific figures. *arXiv preprint arXiv:2110.11624*.

Vera Demberg Iza Skrjanec, Muhammad Salman Edhi. 2022. Barch: An english dataset of bar chart summaries.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. *arXiv preprint arXiv:2005.10433*.

Emiel Krahmer and Kees Van Deemter. 2012. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218.

Joël Legrand, Michael Auli, and Ronan Collobert. 2016. Neural network-based word alignment through score aggregation. *arXiv preprint arXiv:1606.09560*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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.

Ziyang Luo, Yadong Xi, Rongsheng Zhang, and Jing Ma. 2022. Vc-gpt: Visual conditioned gpt for end-to-end generative vision-and-language pre-training. *arXiv preprint arXiv:2201.12723*.

Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. *arXiv preprint arXiv:2010.09142*.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. *arXiv preprint arXiv:2002.12328*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. 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. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Chinmayee Rane, Seshasayee Mahadevan Subramanya, Devi Sandeep Endluri, Jian Wu, and C. Lee Giles. 2021. Chartreader: Automatic parsing of barplots. In *2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 318–325.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yu-An Wang and Yun-Nung Chen. 2020. What do position embeddings learn? an empirical study of pretrained language model positional encoding. *arXiv preprint arXiv:2010.04903*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

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

Jiawen Zhu, Jinye Ran, Roy Ka-wei Lee, Kenny Choo, and Zhi Li. 2021. Autochart: A dataset for chart-to-text generation task. *arXiv preprint arXiv:2108.06897*.