With the support of the
Erasmus+ Programme
of the European Union

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

UNIVERSITÄT
DES
SAARLANDES

MASTER THESIS

Saad Obaid ul Islam

Tackling Hallucinations in Chart Summarization

Institute of Formal and Applied Linguistics, Charles University;
Department of Language Science and Technology, Saarland University

Supervisor of the master thesis:	Mgr. Ondřej Dušek, Ph.D, Prof. Dr. Vera Demberg
Study programme:	Computer Science
Study branch:	Language Technologies and Computational Linguistics

Prague 2022

Declaration

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In Saarbrücken date 19.12.2022

Author's signature

First and foremost, I would like to thank my advisors, Dr. Vera Demberg, Dr. Ondrej Dusek, and Iza Skrjanec, for their guidance and patience. Especially Iza, who I bugged for every small problem. I know I am a nuisance to work with so thank you for being patient with me.

I would like to thank Dr. Marketa Lopatkova and Ms. Bobbye Pernice for helping me during the time I was diagnosed with ADHD. If it were not for their support and kindness in the first year, I would never have finished my studies.

I would like to thank all the friends I made during the LCT program - Niyati, Michael, Anna, Alisa, Claesia, and Allison, for tolerating my goofiness. I would also like to thank my friends back home - Sohail, Mansoor, Javaria, Sarum and Faisal for spending time with me on discord during the lock downs. Special thanks to my friend Mehroo for her moral support and for proof-reading.

Most importantly, I would like to thank my parents - Obaid and Lubna for their encouragement, and patience.

Lastly, I would like to show my gratitude to Erasmus Mundus Joint Masters Programme, co-funded by the Erasmus+ Programme of the EU, for awarding me Erasmus Mundus Scholarship and allowing me to be part of this experience.

Title: Tackling Hallucinations in Chart Summarization

Author: Saad Obaid ul Islam

Faculty of Mathematics and Computer Science: Institute of Formal and Applied Linguistics, Charles University;
Department of Language Science and Technology, Saarland University

Supervisor: Mgr. Ondřej Dušek, Ph.D, Prof. Dr. Vera Demberg

Abstract: Information visualizations like bar charts, line charts, and pie charts are a common way of communicating quantitative data. They are used to get important insights and make well informed decisions. Automatic Chart Summarization is the task to explain and summarize the key takeaways from the chart. Like other natural language generation (NLG) systems, chart summarization systems suffer from a phenomenon called hallucinations. Hallucinations occur when the system generates text that is not grounded in the input. In this research work, we try to tackle the problem of hallucinations in chart summarization. Our analysis shows that a lot of additional information is present in the training data that leads to hallucinations during inference. We also found out that reducing long distance dependencies and addition of chart related information like title and legends improve the overall performance of the system. Furthermore, we propose a natural language inference (NLI) based method to clean the training data and show that our method produces faithful summaries.

Keywords: Natural Language Generation, Data-to-text, Transformers, Information Visualization, Deep Learning, Automatic Summarization

Contents

1. Introduction

Automatic Chart Summarization is becoming an increasingly popular area of research in machine learning community in recent years [Obeid and Hoque, 2020, Hsu et al., 2021, Zhu et al., 2021, Škrjanec et al., 2022, Kanthara et al., 2022], mostly due to the availability of large pre-trained transformer [Vaswani et al., 2017] language models like BERT [Devlin et al., 2018],GPT-2 [Radford et al., 2019], BART [Lewis et al., 2019] and T5 [Raffel et al., 2019].

The large pre-trained transformers are able to produce fluent and coherent text on tasks like machine translation, automatic summarization, natural language inference, sentence similarity, and so on.

The task of summarization comes under the umbrella of natural language generation (NLG) [Paris et al., 2013]. NLG systems produce text given some linguistic or non-linguistic input [Reiter and Dale, 1997]. A common problem in NLG systems, especially when the system consists of a large transformer model, is that the system starts to produce bland, incoherent, and repetitive text. Researchers started to call this problem as *hallucinations* [Koehn and Knowles, 2017, Raunak et al., 2021]. Hallucinations are a concern because it hinders the performance of the NLG system in real world applications. Many efforts have been done to reduce hallucinations in tasks like automatic summarization [Huang et al., 2021], machine translation [Lee et al., 2019] and data-to-text [Rebuffel et al., 2022]. In this thesis, we aim to study hallucinations in the chart-summarization systems and datasets. Furthermore, we provide methods to reduce or eliminate hallucinations.

This research will use the power of state of the art, large pre-trained language models to reduce hallucinations in the current chart summarization systems with some minor pre-processing and post-processing steps.

The rest of the introductory chapter is structured as follows; first, we state the contributions we have made in this thesis. Secondly, we outline the structure of the thesis and briefly talk about what each chapter is about.

1.1 Contributions

This thesis makes the following contributions

- Investigate the reasons for hallucinations in the chart summarization task by analyzing chart summarization datasets, and the models released with these datasets.
- Show the importance of providing more context to the models, and reducing long-distance dependencies in the linearized input format.
- Propose a new training task that reduces hallucinations, perform ablation studies, and conduct human evaluation.

1.2 Outline of Thesis

In the first two chapters (Chapter 2-3) we talk about the background knowledge and related work to understand chart summarization task, and the problem of hallucinations. In Chapter 4, we analyze hallucinations and formulate two hypotheses. In Chapter 5, we conduct experiments to check the correctness of our hypotheses and in Chapter 6, we propose a pre-processing step and training strategy that helps in reducing hallucinations, perform ablation studies, and conduct human evaluation. Lastly, in Chapter 7, we conclude the thesis and outline future research.

2. Background

This chapter will give the background required to understand the chart summarization task and hallucination problem. In Section 2.1 we give a brief overview of natural language generation (NLG), the type of NLG we are interested in i.e. data-to-text, and introduce hallucinations and hallucination types in NLG. In Section 2.2 we introduce the chart summarization task. Section 2.3 will explain sequence-to-sequence architectures and transformer models. This section assumes that the reader has a basic understanding of artificial neural networks Rumelhart et al. [1985] and activation functions [Sharma et al., 2017]. For the background of these concepts, we refer the reader to Goodfellow et al. [2016]. In the last section, we talk about automatic NLG evaluation metrics (Section 2.4).

2.1 Natural Language Generation

Natural Language Generation (NLG) is a branch of artificial intelligence that is concerned with producing understandable text for linguistic or non-linguistic input. Tasks like machine translation, automatic summarization of documents, dialog generation, and image captioning, are all instances of NLG.

2.1.1 Components of NLG

The task of generating natural language is divided into different stages. Reiter and Dale [1997] break it down into the following steps:

- **Content Determination** is the process of determining what information should be included in the text.
- **Discourse Planning** is the process of structuring the set of messages to be conveyed. This is important because good structure and order of the text can make it easier to read and understand.
- **Sentence Aggregation** is the process of combining information into a group of sentences.
- **Lexicalization** involves choosing different domain specific words that will express the input appropriately.
- **Referring Expression Generation** is a step closely related to lexicalization. The difference between the two steps is that expression generation is concerned with generating appropriate noun phrases [1].
- **Linguistic Realisation** is the process of applying grammar rules that makes a text morphologically and syntactically correct.

Before neural networks, combination of above mentioned components were used to build an NLG system. According to Reiter and Dale [1997], the most common architecture consisted of three steps; **text planning**, which combined content determination and discourse

[1] Referring Expression is a noun phrase whose function is to identify individual objects.

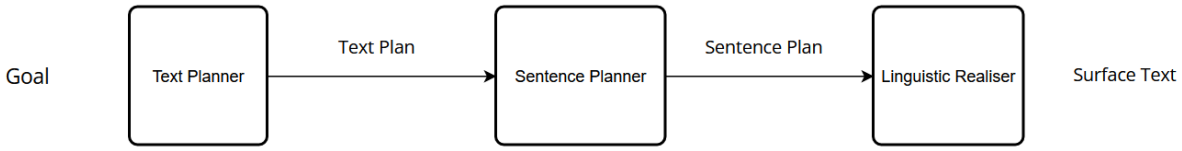


Figure 2.1: Three stage NLG architecture

planning, **sentence planning** combines sentence aggregation, lexicalization, and referring expression generation, and **linguistic realisation** that generated syntactically correct text. Figure 3.2 shows a flowchart of how such architecture would work.

Recent advances in NLG are due to large pre-trained language models like GPT-2 [Radford et al., 2019], T5 [Raffel et al., 2019], and BART [Lewis et al., 2019]. These large language models are trained in an end-to-end fashion and their architectures blur the modules in the architecture used in the pre-neural era.

2.1.2 Data-to-text

A classic problem in NLG is to explain or summarize structured data like spreadsheets, XML files, or databases. Generating texts for sports scorecards [Barzilay and Lapata, 2005], weather forecasts [Liang et al., 2009], and Wikipedia biography tables [Lebret et al., 2016] are some of the example applications that model the NLG task as data-to-text. The input for the NLG system is a linearized input format of the structured data and the output is a text. Traditional method for data-to-text was to implement a kind of pipeline similar to figure 3.2. Neural models like T5 and BART have achieved state-of-the-art (SotA) results on datasets like WebNLG [Gardent et al., 2017], multiwoz [Budzianowski et al., 2018], and ToTTo [Parikh et al., 2020].

2.1.3 Hallucinations

In the context of NLG, hallucination means generating text that is unfaithful to the input text. Formally, Maynez et al. [2020] define hallucinations for automatic summarization task as following:

> A summary S of a document D contains a factual hallucination if it contains information not found in D that is factually correct.

. According to Ji et al. [2022], there are mainly two types of hallucinations:

- **Intrinsic Hallucination**: Generated output that contradicts the source content.
- **Extrinsic Hallucination**: Generated output that cannot be verified from the source content.

If we look at the Figure 2.3, *91 billion* is a type of intrinsic hallucination because it contradicts the input data table that says *30.51 billion*. The second summary is an example of extrinsic hallucination because we cannot confirm from the data if the statement, *"General Electric Company is an American multinational conglomerate founded in 1892"*, is true or not. There is no such information available in the data.

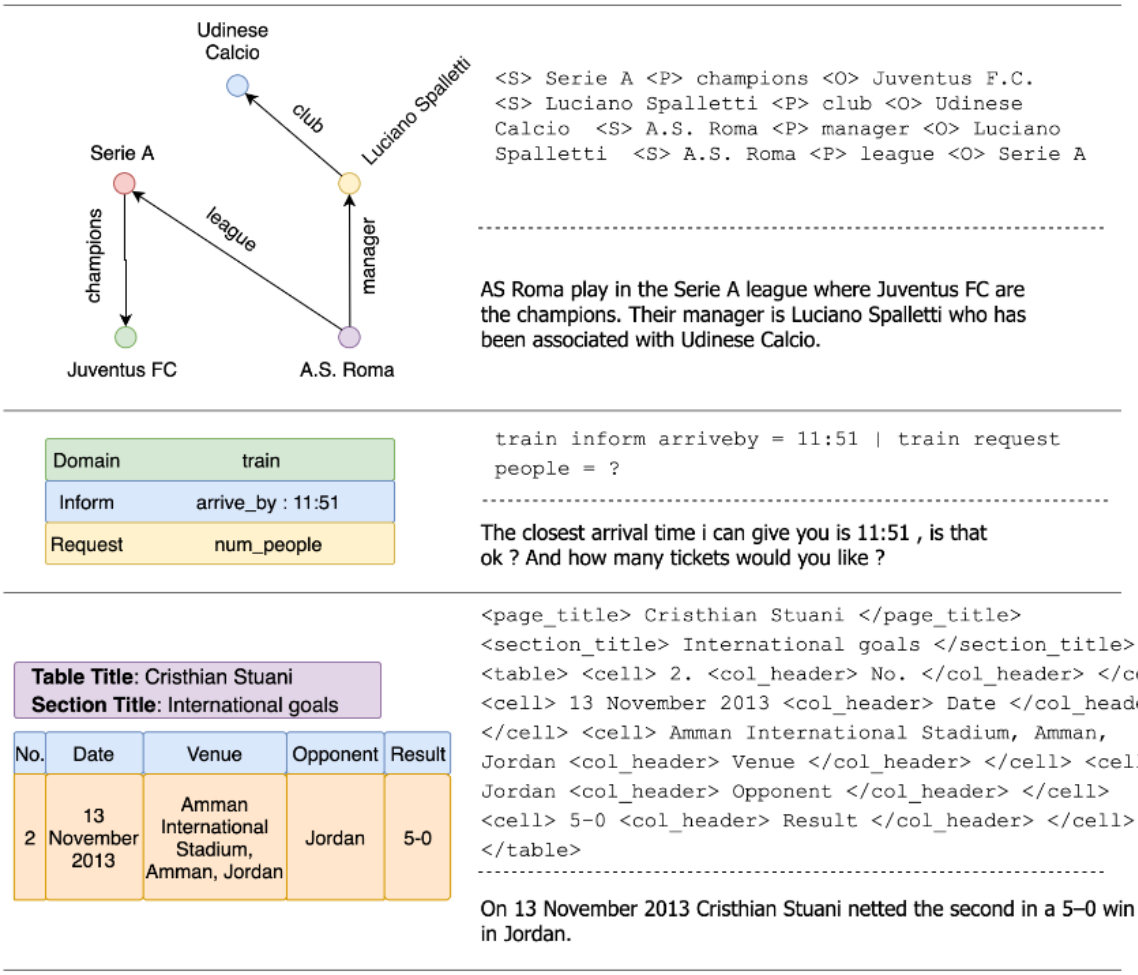


Figure 2.2: Examples from WebNLG, Multiwoz, and ToTTo. Each example consists of the original structured data, their linearized input format and the corresponding reference text [Kale and Rastogi, 2020]

.

Terminology Clarification

Several terminologies are associated with the concept of hallucinations. Commonly used terminologies are *hallucination*, *faithfulness*, and *factuality*. Faithfulness is defined as staying true to the input. It is the opposite of hallucination. So our work in this thesis focuses on maximizing faithfulness, thus minimizing hallucinations.

Another term used when talking about hallucinations is factuality. Factuality refers to the text being based on a fact from the input source. It can be used interchangeably with faithfulness.

2.2 Chart Summarization

Information visualizations or charts are used by the scientific and business community to present complex data in a neat and informative manner. Automatic summarization of charts or chart-to-text is the task of summarizing/describing key insights and takeaways from a chart into natural language. Figure 2.4 shows an example of a chart, its underlying data,

Segment	Value of assets in billion U.S. dollars
Renewable energy	15.94
Power	26.73
Healthcare	30.51
Aviation	41.65
Capital	117.55

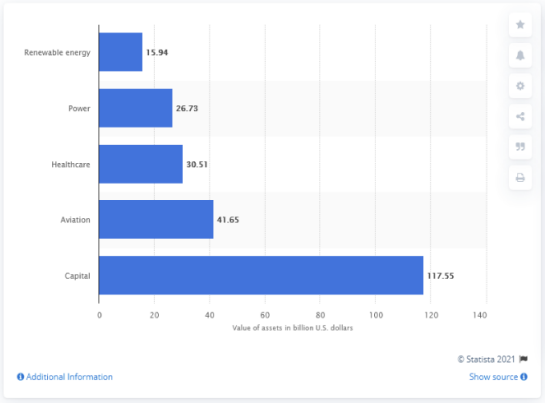
Title: General Electric 's total assets in FY 2019 , by segment (in billion U.S. dollars)

This statistic represents General Electric 's total assets in the fiscal year of 2019 , with a breakdown by segment . In its healthcare segment , the company had assets to the value of around 91 billion U.S. dollars

This statistic represents General Electric 's total assets in the fiscal year of 2019 , with a breakdown by segment . In its healthcare segment , the company had assets to the value of around 30 billion U.S. dollars . General Electric Company is an American multinational conglomerate founded in 1892.

Figure 2.3: Data and two summaries. Red indicates hallucination [Kale and Rastogi, 2020] .

and the corresponding summary or description of the chart.



Segment	Value of assets in billion U.S. dollars
Renewable energy	15.94
Power	26.73
Healthcare	30.51
Aviation	41.65
Capital	117.55

Title: General Electric 's total assets in FY 2019 , by segment (in billion U.S. dollars)

This statistic represents General Electric 's total assets in the fiscal year of 2019 , with a breakdown by segment . In its healthcare segment , the company had assets to the value of around 30.5 billion U.S. dollars .

Figure 2.4: Example of chart, underlying table, and chart summary [Kanthara et al., 2022] .

The chart-to-text generation task can be modelled in two ways; image-to-text and data-to-text. In an image-to-text system, the model takes in the chart as input and produces a text related to that chart. A number of image-to-text systems for information visualization have been developed using neural models [Chen et al., 2020a, Qian et al., 2021, Hsu et al., 2021].

If the task is modelled as data-to-text, the model takes in the underlying data of the chart and produces text. Datasets and approaches proposed by Obeid and Hoque [2020], Kanthara et al. [2022], and Škrjanec et al. [2022] model chart summarization task as data-to-text.

For this thesis, we will model the chart summarization task as **data-to-text**.

2.3 Neural Architectures

2.3.1 Recurrent Neural Network

Recurrent Neural Network (RNN) was first introduced in Rumelhart et al. [1985] as hopfield networks. This type of neural network consists of a hidden state h that gets updated on variable length input sequences $x = (x_1, x_2, ..., x_T)$ and generates an output y. At time step t, the hidden state h_t of RNN is updated as follows

$$h_t = f(h_{t-1}, x_t) \tag{2.1}$$

where f is a non-linear activation function.

2.3.2 Encoder-Decoder Architecture

Cho et al. [2014] introduced a novel neural network architecture that learns to *encode* a sequence of variable length into a fixed-length vector. This fixed-length vector is then *decoded* back to variable length. The encoder is an RNN which reads each input sequence and updates the hidden representation using equation 2.1. The hidden state of the RNN-Encoder is the summary $\boldsymbol{c}$ of the whole input sequence. The decoder is another RNN which is trained to generate the output sequence by predicting the next symbol y_t. The difference between decoder RNN hidden state h_t is that it is conditioned on the summary $\mathbf{c}$, along with output generated at previous timestamp y_{t-1}. At time step t, the decoder hidden state is then computed as follows

$$h_t = f(h_{t-1}, y_t, \mathbf{c}) \tag{2.2}$$

The two components; Encoder and Decoder, are then jointly trained. Once the model is trained, the model can be used to generate a target sequence given a source sequence.

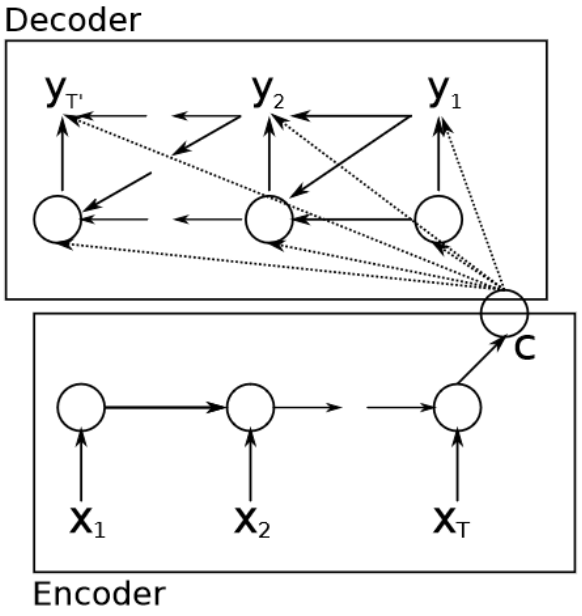


Figure 2.5: Illustration of Encoder-Decoder architecture [Cho et al., 2014]

The problem with RNN based Encoder-Decoder model is that the information needs to be compressed and during that process some information is lost. Especially, information that is found earlier in the sequence. This problem can be remedied by using bi-directional RNNs but they are only good for shorter sequences and the problem is preserved for longer sequences because of the vanishing gradient problem.

2.3.3 Transformers

Attention

Bahdanau et al. [2014] introduced attention between encoder-decoder blocks to fix the information problem caused by the encoder-decoder architecture. Given previous RNN decoder state s_{t-1} and encoder state h_i, Bahdanau et al. [2014] attention is computed in the following way:

- **Attention energy** or alignments is denoted by a function called *alignment* which is a feed-forward neural network. It measures how well the inputs around position i match with the output at position t. At time t, it computes energies $e_{t,j}$ given encoder state h_i and decoder state s_{t-1}

$$e_{t,i} = \text{alignment}(s_{t-1}, h_i) \tag{2.3}$$

- **Attention distribution** or weights $\alpha_{t,i}$ is computed by applying softmax function to $e_{t,j}$

$$\alpha_{t,i} = \text{softmax}(e_{t,j}) = \frac{\exp(e_{t,j})}{\sum_{k=1}^{I_x} \exp(e_{ik})} \tag{2.4}$$

  Where I_x is the set of encoder hidden states.

- **Context vector** is similar to the summary **c** in Section 2.3.2. However, it is computed as the weighted sum of all hidden encoder state I. This context vector c_t is then fed into the decoder at each time step.

$$c_t = \sum_{i=1}^{I_x} \alpha_{t,i} h_i \tag{2.5}$$

In summary, the first step in the attention mechanism is to compute matching scores between inputs and outputs. The second step is to generate weights using the softmax function and the third step is to compute the attention context vector as the weighted sum of all encoder hidden states. Figure 2.6 shows the flow of the attention mechanism. This type of attention is also called concatenating/additive attention.

Self-Attention and the Development of Transformers

As the name suggests, self-attention allows an encoder to attend to other parts of the input during processing. Attention within an encoder or decoder block was first introduced as shallow attention fusion by Cheng et al. [2016]. However, later on, [Vaswani et al., 2017] coined the term self-attenstatesand used it extensively in their proposed architecture called the transformer. The transformer model is a type of neural network that relies entirely on (self-)attention mechanism. While the transformer architecture does not use RNNs, it

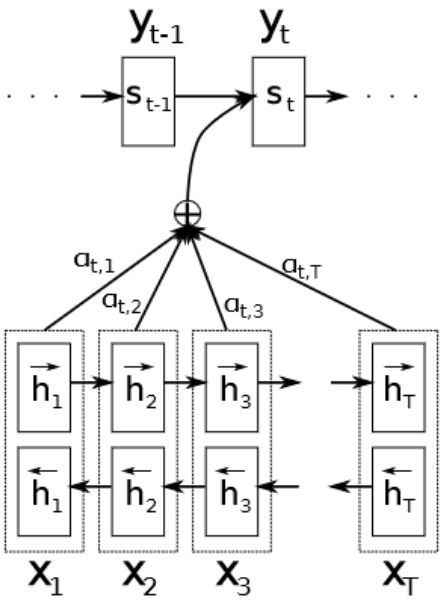


Figure 2.6: Illustration of encoder-decoder architecture with attention [Bahdanau et al., 2014]

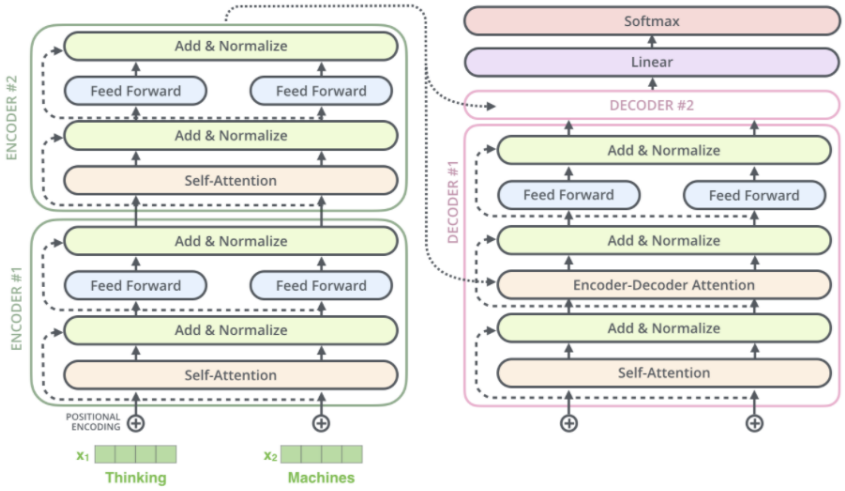


Figure 2.7: Illustration of transformer by Alammar [2018]

retains the encoder-decoder architecture proposed by Cho et al. [2014]. Figure 2.7[2] shows the encoder-decoder blocks of the proposed transformer architecture.

The encoder block of a transformer model is made up of N stacked identical layers. Each layer has two sub-layers, The first sub-layer computes self-attention and the second sub-layer is a feed-forward network (FFN). The decoder block is also composed of stacked decoder layers, which consists of two sub-layers present in the encoder, plus an additional sub-layer that computes attention over the output of the encoder stack or vanilla attention (section 2.3.2).

The self-attention in the transformer model is defined as the mapping of a query vector and a set of key-value vector pairs to an output vector. Given a sequence of n words

[2]https://jalammar.github.io/illustrated-transformer/

represented as a vector X, the attention, for a set of queries Q of dimension d_k, keys K of dimension d_k, and values V of dimension d_v, is defined as

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.6}$$

where queries, keys, and values are computed from the input word representation using linear transformations

$$Q = XW^Q \tag{2.7}$$

$$K = XW^K \tag{2.8}$$

$$V = XW^V \tag{2.9}$$

for trainable weights W_Q, W_K and W_V. Figure 2.8 show the flow of operations of self-attention.

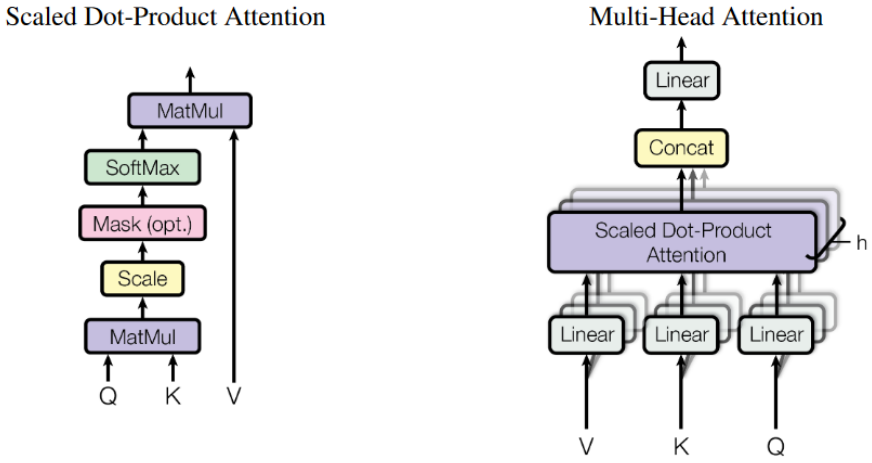


Figure 2.8: Illustration of Scaled dot-product attention (self-attention) and multi-head attention. [Vaswani et al., 2017]

Vaswani et al. [2017] found that there are multiple different aspects a sequence element wants to attend to and a single weighted-average is not sufficient. So they extended self-attention to multi-headed self attention. The multi-headed attention is defined as follows

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, ..., head_h)W^O \tag{2.10}$$

where

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.11}$$

The birth of transformers gave rise to a series of large pre-trained language models like BERT, GPT-2, BART, and T5 which employed self-attention with different learning techniques to advance the field of natural language processing.

2.3.4 Text-to-Text Transformers

Preliminary: Modeling Objectives

- **Causal modeling** or language modeling is the training method/objective used when the target is to predict the next token given the history.
- **Masked Language modeling** is the training method/objective used when the target is to fill the blank (mask) token, given the left and right context.

Preliminary: Models

- **GPT-2** Radford et al. [2019] is a transformer model that was developed by researchers at OpenAI. It consists only of stacked decoder layers and is trained in a causal manner on a large amount of text corpus. For downstream tasks, it is then fine-tuned on a small dataset. GPT-2 is good for text generation tasks.
- **BERT** [Devlin et al., 2018] is a transformer model that was introduced by researchers at Google. It was trained on a large amount of text corpus and then later fine-tuned on downstream tasks. It consists of only stacked encoder layers and employs bidirectionality in the training process by using masked language modeling. BERT is good for text classification tasks.

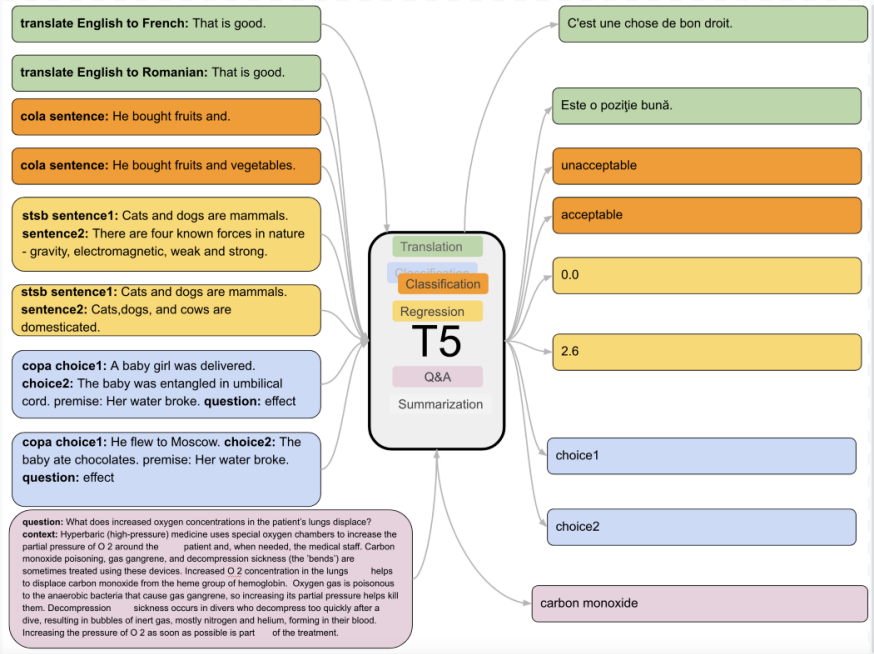


Figure 2.9: A single T5 model for multiple tasks [Rajasekharan, 2019].

T5: Text-To-Text Transfer Transformer

The researchers at Google proposed a model called T5, which in theory, can solve any type of NLP problems if that problem is converted to text-to-text format. Unlike BERT and GPT-2, Raffel et al. [2019] employed the exact same encoder-decoder architecture as the original transformer (Figure 2.7) and combined causal and masked language modeling. The encoder

was trained with the masked language modeling objective and the decoder was trained with the causal objective. This type of multi-objective training allows the model to solve any NLP task as long as it is presented as text-to-text.

The authors also implemented multi-task learning, where they trained a single model on multiple tasks by appending prefix. The tasks included machine translation, summarization, question answering, and so on. To give an example, when the model is asked to translate the sentence "That is good." from English to German, the model would be fed the sequence "translate English to German: That is good." and would be trained to output "Das ist gut." Task specific prefix enables a single model to perform several tasks. Figure 2.9[3] shows how one model can be prompted to produce output for several tasks.

2.4 Evaluation Metrics in NLG

- **Precision** is the fraction of true instances among retrieved instances.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{2.12}$$

- **Recall** is the fraction of true instances that were retrieved.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{2.13}$$

It is well known in the NLP community that evaluation of NLG models is hard [Howcroft et al., 2020, van Miltenburg et al., 2021]. Factors like quality, fluency, verbosity, and consistency need to be considered when evaluating NLG models. Compare this to a text classification system, which only requires accuracy for evaluation. Typically, NLG models are evaluated across multiple automatic and manual evaluation metrics. The two most common metrics for NLG are BLEU and ROUGE. Nowadays, perplexity of a model is also considered to measure fluency.

- **BLEU**: Bilingual Evaluation Understudy, commonly known as BLEU, measures n-gram precision between reference and candidate/hypothesis text. Normally, the value of n is considered to be 4. It was proposed by **?** to evaluate machine translation systems. Since then, it has been used to evaluate NLG like automatic summarization and data-to-text. BLEU is computed as follows:

$$\text{BLEU} = \text{B.P} \cdot \exp(\sum_{n=1}^{N} w_n \log_e p_n) \tag{2.14}$$

  where p_n is precision of n-gram, w_n are uniform weights, and $B.P$ is brevity penalty which penalises sentences that are short. $B.P$ is defined as

$$B.P = \begin{cases} 1 & \text{if c} > \text{r} \\ \exp(1 - \frac{r}{c} & \text{if c} \leq \text{r} \end{cases} \tag{2.15}$$

  where c is the number of unigrams in candidate sentences and r is the best match length for each candidate sentence in the corpus. The problem with BLEU is that it does not capture the quality of the text and only measures surface overlap.

[3] https://towardsdatascience.com/t5-a-model-that-explores-the-limits-of-transfer-learning-fb29844890b7

- **ROUGE**: Recall Oriented Understudy for Gisting Evaluation, commonly known as ROUGE, measures n-gram recall between reference and candidate/hypothesis text. Normally, the value of n is considered to be 1 and 2. It was proposed by Lin [2004] to evaluate summarization systems. ROUGE has the same problem as BLEU. ROUGE is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{\text{S}\in\text{ReferenceSummaries}} \sum_{\text{gram}_n\in\text{S}} Count_{\text{match}}(gram_n)}{\sum_{\text{S}\in\text{ReferenceSummaries}} \sum_{\text{gram}_n\in\text{S}} Count(gram_n)} \tag{2.16}$$

  where n is length of n-gram, $gram_n$ and $Count_{match}gram_n$ is the maximum number of n-grams co-occuring in candidate and a set of references.

- **Perplexity**: Perplexity (PPL) is defined as exponentiation of negative-log likelihood of a sequence. Given a tokenized sequence $X = x_0, x_1..., x_t$, perplexity is computed as follows:

$$\text{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(\frac{x_i}{x_{<i}})\right\} \tag{2.17}$$

  Intuitively, the perplexity of an NLG system indicates its ability to predict uniformly among a set of specified tokens in a corpus. Perplexity is also equivalent to the exponentiation of cross entropy between data and model predictions. It is also informative for measuring grammaticality when evaluated for a large pre-trained language model like GPT-2.

Due to the rise of large scale pre-trained transformers, there is a new breed of metrics like bertscore [Zhang et al., 2019] and bleurt [Sellam et al., 2020]. These metrics are called learned metrics. Transformers like BERT are fine-tuned for producing ratings. When used for evaluation, they produce a score that tells how much the candidate/hypothesis conveys the meaning of the source.

BLEURT

Sellam et al. [2020] introduced a novel metric that assigns ratings to candidate text when compared to reference text. This metric is inspired from human evaluation where generated text is presented to the annotators and they are asked to rate the quality and meaning of the text. Figure 2.10 shows how annotators are asked to evaluate output text.

Figure 2.10: Example questionnaire from [Sellam et al., 2020].

Human beings are still unrivaled when it comes to assessing the quality of text. However, human evaluation can take weeks to finish which interferes with the development workflow. BLEURT was introduced as a low latency proxy to human beings.

BLEURT can capture non-trivial semantic similarities between two texts. It is trained on collection of ratings; WMT shared task evaluation dataset [Ma et al., 2018]. Additionally, it was pre-trained on synthetic data as well in order to learn how to rate a wide variety of tasks. Optionally, BLEURT can be further fine-tuned on task specific ratings. The training flow for developing BLEURT is shown in figure 2.11:

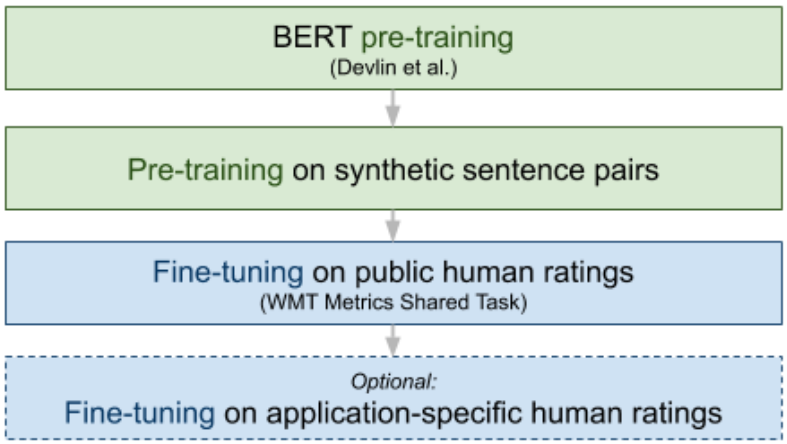


Figure 2.11: An example questionnaire from Google blog.

Figure 2.12 shows how BLEURT evaluates candidate and reference text.[4]

Figure 2.12: BLEURT evaluating candidate and reference text. The transformer is a BERT model pre-trained on ratings data.

Given a candidate-reference pair, BLEURT assigns are score between $[-1, 1]$. -1 being the lowest score and 1 being the highest.

NUBIA

NUBIA [Kane et al., 2020] is an interpretable metric that utilizes three machine learning models to produce a score. NUBIA metric consists of three components:

1. **Feature extractor** extracts features from reference-candidate pairs like semantic similarity, logical entailment, and grammaticality. Following models are used to extract these features:

   (a) RoBERTa-STS large [Liu et al., 2019] is used to get sentence similarity score. A good candidate should have a high similarity score with the reference.

[4]Paper talk: https://papertalk.org/papertalks/6651

(b) RoBERTa-NLI large [Liu et al., 2019] is used to get a classification score of either 0 (contradiction), 1 (undecided/neutral) or 2 (entailment). The rationale for using NLI is that a good candidate text will convey the core meaning and argument of the reference text.

(c) GPT-2 [Radford et al., 2019] to capture the grammaticality of the candidate text. The feature extracted using this model is perplexity score.

2. **Aggregator** is a fully connected neural network that is trained to approximate a function that maps the above mentioned features to a quality score that reflects how interchangeable the candidate and reference text are.

3. **Calibrator** scales the raw scores of the aggregator between 0 and 1.

The whole process of feature extraction, aggregation, and calibration is inspired from the process of human evaluating a source candidate-sentence. Here, you can assume that the aggregator is a human that is looking at several features when assigning a score between 1 and 100. Figure 2.13 shows how a score is assigned to the candidate-reference pair using NUBIA.

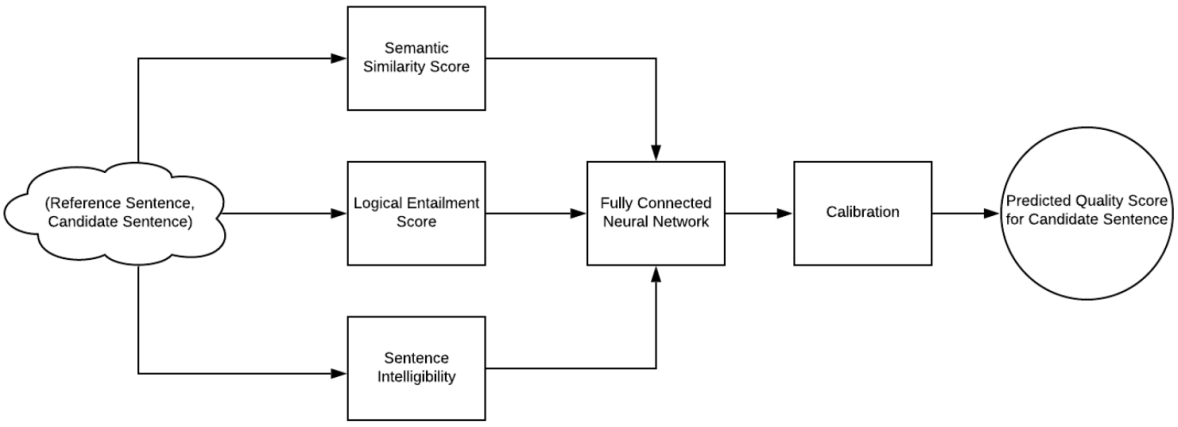


Figure 2.13: Full workflow of NUBIA [Kane et al., 2020].

The reason we use this metric is interesting is that NUBIA produces a final score, as well as individual semantic similarity, entailment, neutrality, contradiction, and grammaticality score. Entailment and contradiction scores are helpful for evaluating faithfulness. At the end of the computation of the score, NUBIA outputs six scores: Logical Agreement (LA), Contradiction (CONTRA), Neutrality (NEUT), Grammatically, Semantic Similarity (SemSim), and NUBIA.

3. Related Work

In this chapter we review the work related to this thesis. We talk in detail about the recent advances in chart summarization (Section 3.1) and how hallucinations in NLG systems are being tackled (Section 3.3). Lastly, we introduce natural language inference (NLI) (Section 3.4) and talk about how NLI is being used to reduce hallucinations.

3.1 Recent work in Chart Summarization

Over the past two years, several chart-summary pair datasets and models have been developed and made available to the public.
Obeid and Hoque [2020] created a dataset crawled from statista.com called Chart-to-Text (C2T). They modelled the chart summarization task as a data-to-text problem and adapted a transformer developed by Gong et al. [2019] for data-to-text generation. To prevent the model from hallucinating, the authors introduced data variable substitution during pre-processing and post-processing. Before training, all the entities in the summaries are substituted with a special tokens and then the transformer model is trained on those special token summaries. During the inference time, the model generates delexicalized summaries which are later lexicalized through the data variable substitution module.

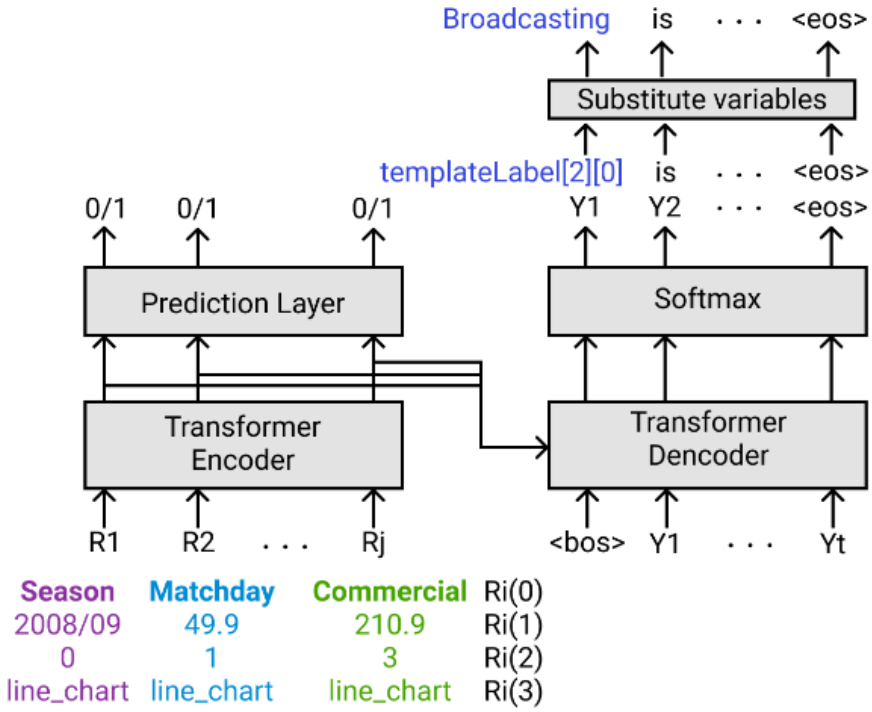


Figure 3.1: The model takes chart data and some metadata as input and generates summary containing data variables that refer values within a data table [Obeid and Hoque, 2020].

Zhu et al. [2021] created a template based dataset of Charts and two analytical summaries called AutoChart. This dataset was specifically developed for modelling chart-to-text as an image-to-text problem. However, the authors published metadata files which contained the underlying chart data which can be utilized if chart-to-text is modelled as data-to-text.

Hsu et al. [2021] created a dataset of scientific figures collected from arXiv along with captions called SciCap. They used the underlying chart data to build a model using an RNN. More recently, Kanthara et al. [2022] released a large dataset also crawled from statista.com called chart-to-text. They tested their dataset on several models like BART and T5 transformers. Barch [Škrjanec et al., 2022] is another dataset that consists of human written bar chart summaries. Each chart has corresponding underlying data and with multiple summaries. Multiple summaries are supposed to simulate real world environments, where more than one summary can be correct for a chart.
Table 3.1 shows the total size of the existing publicly available datasets.

Dataset	Total	Train	Validation	Test
Chart-to-Text [Obeid and Hoque, 2020]	8,147	5,703	1,222	1,222
Autochart [Zhu et al., 2021]	23,543			
SciCap [Hsu et al., 2021]	2,000,000	1,600,000	170,000	170,000
Chart-to-Text [Kanthara et al., 2022]	34,811	24,367	5,222	5,222
Barch [Škrjanec et al., 2022]	1,063	660	213	190

Table 3.1: Summary of the dataset sizes and train-val-test splits. Autochart dataset has no splits.

3.2 Datasets of Interest

As mentioned in the last section, we mainly have three chart summarization datasets that are designed for data-to-text modelling. We also mentioned that the autochart dataset [Zhu et al., 2021] can be used for data-to-text because the authors published a metadata file that contains chart data.

From Table 3.1, we will be using datasets (1), (2), and (4).

3.2.1 Chart-to-text dataset by [Obeid and Hoque, 2020] (c2t-small)

This dataset contains simple and complex, line, and bar charts. A simple bar chart contains a set of bars and a simple line plot contains a single line. Complex bar charts contain stacked bars and complex line plots contain more than one line. The ratio between line and bar charts is almost 1 : 1.

Statistic	Value
Mean Token Count/Summary	113.4
Mean Sentence Count/Summary	5.2
Vocab Size	19,150
Total Tokens	941.8K

Table 3.2: Dataset statistics [Obeid and Hoque, 2020]

The discourse structure of the summaries is mainly the same throughout. Summaries do not mention the type of chart. The first sentence introduces the chart by mentioning the title. The following sentence(s) mention some notable features like the highest/lowest value, trend, or the first and last value.

3.2.2 Chart-to-text dataset by [Kanthara et al., 2022] (c2t-big)

This dataset contains simple and complex, line, and bar charts, and pie charts. However, about 87% of the dataset contains bar charts followed by 10.2% of line plots.

Statistic	Value
Mean Token Count/Summary	53.65
Mean Sentence Count/Summary	2.59
Vocab Size	57,812
Total Tokens	1.4mil

Table 3.3: Dataset statistics [Kanthara et al., 2022]

The discourse structure of the summaries is mainly the same as c2t-small (see Subsection 3.2.1).

3.2.3 Autochart dataset by [Zhu et al., 2021]

This dataset contains bar, line, and scatter plots generated from a template. Each chart has two summaries to simulate a real world environment where more than one summary can be correct. There are a total of 10,232 charts with 23,543 summaries.

Statistic	Value
Mean Token Count/Summary	140
Mean Sentence Count/Summary	8

Table 3.4: Dataset statistics [Zhu et al., 2021]

As the entire dataset is generated using a template, including the summaries, all the summaries follow a pattern. In the first sentence the summary mentions the type of chart and the title of the chart, in the following sentences it talks about what x and y axis represent and then talks about the trends and the highest lowest values.

The template for summaries use information like bounding boxes of coordinates to figure out a trend. This information is present in the meta-data file along with other chart information like title, x-y axis labels, x values, y values, and image index.

From the datasets mentioned in table 3.1, (1) (4) and (5) are designed for data-to-text task[1] so we will consider them in this thesis. Table 3.5 shows the models we are interested in and the ones we will be comparing our results to.

Model	BLEU	BLEURT	PPL
Obeid and Hoque [2020] Transformer	18.54	-	-
Kanthara et al. [2022] BART	36.36	0.12	12.55
Kanthara et al. [2022] T5	37.01	0.15	10.00

Table 3.5: Results of the models we are interested in. PPL is perplexity.

[1]They can be used for image-to-text task as well.

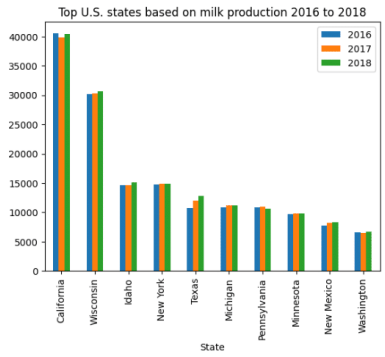
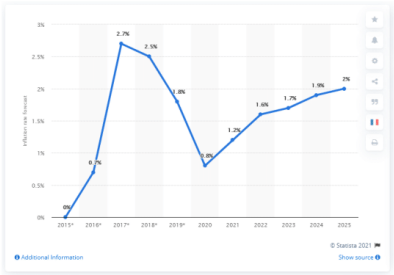
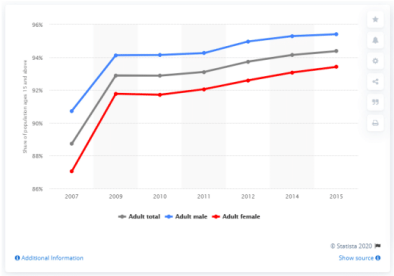
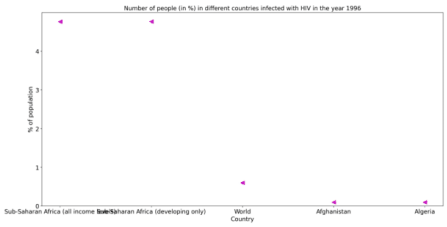
Index	Dataset	Chart	Summary
1	c2t-small		This statistic shows the ten U.S. states with the highest amount of milk production from 2016 to 2018 . California , was the leading producer , where 40.4 billion pounds of milk were produced in 2018 . Milk production Dairy farming is an agricultural business which is engaged in the long-term milk production within the dairy industry .
2	c2t-big		As of November 2020 , the inflation rate of the Consumer Price Index is expected to be 0.8 percent throughout 2020 , before rising to 1.2 percent in 2021 , and 1.6 percent in 2022 . During the provided time period the inflation rate was at it 's highest in 2017 when it reached 2.7 percent .
3	c2t-big		As of 2015 , South Africa 's total literacy rate was around 94.37 percent , which means almost 95 percent of all South Africans could read and write .
4	Autochart		The scatter plot shows the number of number of people (in %) in different countries infected with hiv in the year 1996 in 4 countries. (Sub-Saharan Africa (all income levels), Sub-Saharan Africa (developing only), World, Afghanistan, Algeria,). The unit of measurement in this graph is % of population as shown on the y-axis. The peak of the number is recorded in Sub-Saharan Africa (developing only) and the lowest number is found in Afghanistan. The number changes can be related to the national policies of the country.

Figure 3.2: Example of three datasets. (1) is a complex bar chart, (2) is a simple line plot, (3) is a complex line plot, and (4) is a scatter plot.

3.3 Causes of Hallucinations

As defined in section 2.1.3, hallucination is the text in the output that is unfaithful to the input. There are three main causes for hallucinations in NLG systems.

- **Source-Reference Divergence**: When building large scale datasets heuristically, it could be possible that the reference text is not entirely supported by the source. For instance, when constructing WIKIBIO [Liu et al., 2017], the authors took the Wikipedia infobox [2] as the source and the first sentence of the Wikipedia page as a target. This can potentially lead to divergence in the source and reference text, as sometimes information provided in the infobox is not present in the first sentence or even the first paragraph of the Wikipedia page. Dhingra et al. [2019] found out that 62% of the first sentences in the WIKIBIO dataset do not have the additional information present in reference text. Another problem is the presence of duplicates. If duplicates are not filtered out, the generated text will be repetitive.

- **Training-modelling Choices**: Parikh et al. [2020] showed that even if a dataset is clean and there is little to no divergence between source-reference text, modelling choices can affect the generated output. Models learn wrong correlations in the training samples as the data and model gets bigger and bigger. Another problem during training is the *Parametric Knowledge Bias* (PKB). Pre-training models on large corpora like Common Crawl results in models learning language in its parameters. This is called parametric knowledge which helps improve the performance when the model is fine-tuned on a downstream task. As good as these pre-trained models are, Longpre et al. [2021] discovered that these models prioritize parametric knowledge over the input knowledge which results in extrinsic hallucinations.

- **Decoding Strategies**: During inference time, the task of the decoder is to generate some string $y*$ according to the given model p using some rule. Those rules refer to the decoding strategies like greedy search, beam search, and sampling techniques. Dziri et al. [2021] illustrate that decoding strategies that improve the diversity and fluency of the output like sampling techniques are correlated with increased hallucinations.

3.3.1 Hallucination Mitigation in Data-to-text

There have been certain strides on mitigating hallucinations in NLG systems, in particular, data-to-text models. At data level, several clean and faithful datasets like ToTTo [Parikh et al., 2020] and RotoWire-FG (fact grounded) [Wang, 2020] have been developed. For ToTTo, authors ensured that targets exclude hallucinations by asking annotaters to revisit existing Wikipedia candidate sentences and remove the parts that were unsupported by the WIKIBIO table. For pre-processing, Nie et al. [2019], utilize a natural language understanding (NLU) module to improve the equivalence between input data and target. At the modelling and decoding level, planning and skeleton generation are common methods to improve faithfulness. Wang et al. [2020] propose a two step generation with separate text planner and sequence generator. First, the text planner predicts the plausible content plan based on input data and in the second step, the sequence generator generates text based on that content plan. AGGGEN (Aggregating while Generation) [Xu et al., 2021] is an end-to-end version of the previously mentioned two step generation that jointly learns to plan

[2] https://en.Wikipedia.org/wiki/Help:Infobox

and generate. To remove hallucinations at decoding level, [Rebuffel et al., 2022] proposed a multi-branch decoder that leverages world level alignment labels between the input data and target text to learn relevant parts.

3.4 Natural Language Inference

The natural language inference (NLI) is the task of determining whether a given hypothesis h is true (entailment), undetermined (neutral) or false (contradiction) for a given premise p[3].

Premise	**Hypothesis**	**Label**
A man inspects the uniform of a figure in some East Asian country.	The man is sleeping.	contradiction
An older and younger man smiling.	Two men are smiling and laughing at the cats playing on the floor.	neutral
A soccer game with multiple males playing.	Some men are playing a sport.	entailment

Table 3.6: Examples of entailment, contradiction, and neutral hypothesis from papers with code.

3.4.1 Zero-shot Classification as Textual Entailment

A zero-shot text classification problem setup is the task of classifying text without having seen any labelled data. Nowadays, this type of setup is usually seen using large scale transformers and is used for making a model to do something that it was not explicitly trained to do. Yin et al. [2019] showed that large pre-trained NLI transformer models are zero-shot sequence classifiers. The idea is to take a sequence we are interested in labeling, as premise p, and turn each candidate label to hypothesis h. If the NLI model predicts that the p entails h, we take the label to be true.

One of the applications of NLI is to use it as a metric. A system that can identify implications of sentences must have a good understanding of how language works [MacCartney, 2009]. To this end, NLI has been used for building semantic search systems, and for evaluating automatic summarization, and question answering systems. We can use zero-shot large pre-trained NLI transformers for evaluating our chart-summarization models.

3.4.2 NLI for Evaluating Faithfulness

NLI is useful for the topic of this thesis because we do not want unfaithful text in our summaries. Falke et al. [2019] utilized a pre-trained entailment based method to assess whether the generated output is entailed in the source or not. For data-to-text in particular, [Dušek and Kasner, 2020] used a transformer fine-tune for NLI to evaluate generated text.

[3]NLI definition and example: https://paperswithcode.com/task/natural-language-inference

The generated text is said to be true if it mentions all and only the input facts. Previously in chapter 2, we also mentioned NUBIA, an automatic metric that utlilizes NLI model to evaluate logical consistency between reference-candidate pair.

4. Problem Identification

In this chapter, we identify the types of hallucinations in the generated summaries and talk about problems with linearized input data (Section 4.1). In Section 4.2, we identify problems in the training data.

4.1 Hallucinations in Generated Summaries

We analyzed the data and focused on what type of hallucinations are often generated for chart summarization. The dataset we used to analyze is $c2t - small$ (see Section 3.2.1) and the generated summaries are from the outputs generated by the transformer model [Obeid and Hoque, 2020]. We categorized the hallucinations in chart summarization task as follows:

- **Entity** (ENT) based hallucinations happen when the model starts generating named entities or data values that are not contained in the chart data. This is the most common type of hallucination.
- **Outside Information** (OI) is the text that might look true but cannot be verified from the chart data. These type of hallucinations are also called extrinsic hallucinations (see Section 2.1.3).

There is a possible overlap between ENT and OI in the sense that named entities can be categorized as OI. The key difference between them is that OI is the generated text learned from the training summaries and ENT is the generated text that is a result of poor training. OI comes under extrinsic hallucinations and ENT comes under intrinsic hallucinations.

We analyzed fifty chart summaries and the hallucination statistics are reported in Figure 4.1. In the Figure 4.2, we see examples of the two types of hallucinations. Furthermore, the generated summaries also had a problem of repetitive and incoherent text.

4.1.1 Input Format is Important

As previously mentioned in Section 3.3, modelling choices are one of the causes of hallucinations. To mitigate the hallucinations caused due to the model, we use the unified text-to-text transfer transformer or T5 (see Section 2.3.4). It has previously been shown that data-to-text problems can be formulated as text-to-text [Kale and Rastogi, 2020]. We think that T5 will give better results on chart-to-text task because unlike the transformer model used by Obeid and Hoque [2020], T5 has been pre-trained on a large corpus. Using T5 should reduce the ENT type hallucinations along with repetitions and incoherence.

The input fed to the model when it comes to data-to-text is a linearized format of the data table. Obeid and Hoque [2020] linearize the data table in a very specific format as shown in Table 4.1. The linearized data table lacks title of the chart and consists of x and y axis labels, values, and chart type. This linearized data table is then used to generate a delexicalized summary which is then lexicalized by the data variable substitution module (see Figure 3.1). Even though this data variable substitution step was put in place to tackle hallucinations and incorporate chart related information like the title of that chart, we observe that it fails to do that.

Pre-trained T5 and BART models trained by Kanthara et al. [2022] (see Table 3.5) produces good results, but we believe that by modifying the input format of the data table, we

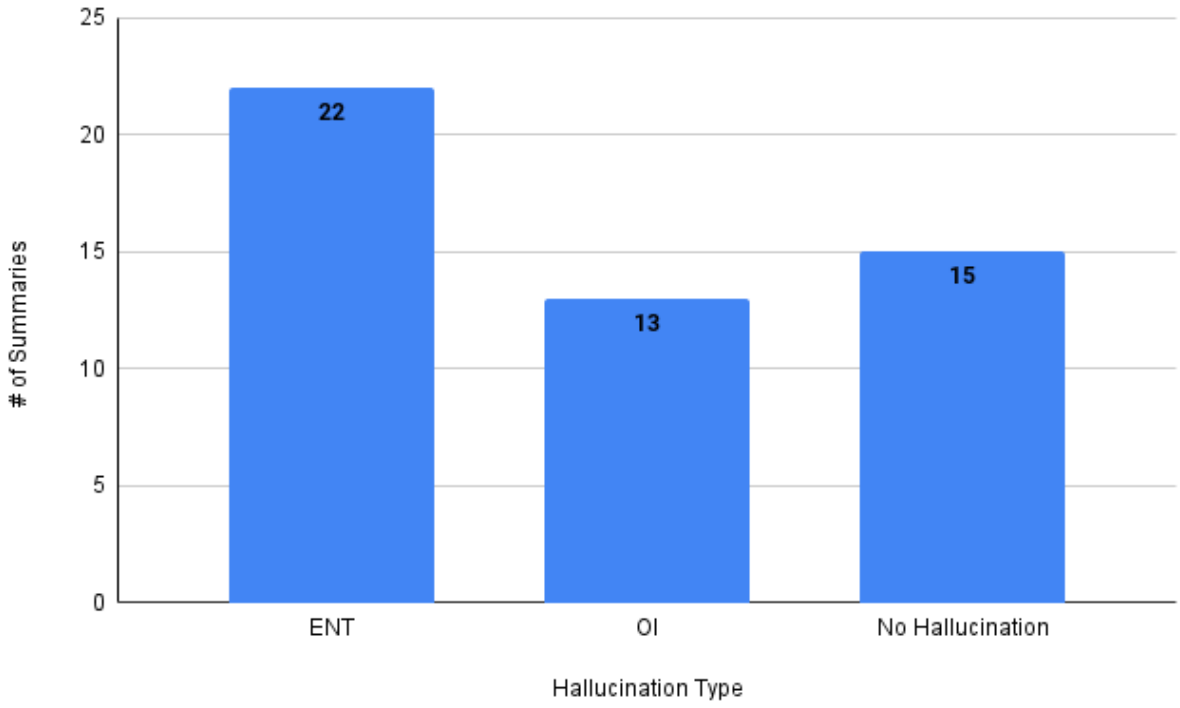


Figure 4.1: Entity (ENT), and Outside Information (OI) hallucination distribution.

Index	Data	Generated Summary
1	title: U.S. Millennials : most popular social network 2016 , by age group Data: 0 1 2 3 4 5 6 Platform Facebook Instagram YouTube LinkedIn Pinterest Snapchat Twitter 18-24 36 24 12 7 9 9 3 25-29 41 17 14 10 8 9 1 30-34 50 13 11 10 8 5 2	This statistic shows the results of a survey on the U.S. Millennials popular in Millennials as of 2018 , sorted by age . During the survey , 36 % of respondents aged 18-24 to years stated that they used Gmail as their primary popular provider . Instagram was ranked second with 24 % .
3	title: University applicants in the United Kingdom (UK) 2010 to 2018 X_Axis['Year']: ['2018', '2017', '2016', '2015', '2014', '2013', '2012', '2011', '2010'] Y_Axis['Number', 'of', 'university', 'applicants']: ['636960', '649700', '674890', '673040', '659030', '635910', '616700', '668150', '658560']	The statistic shows the United of University produced in United Kingdom from 2010 to 2018 . In 2018 , the company spent some 636960 million British pounds in Number university . United Kingdom is the world 's largest cosmetics and beauty company , concentrating on hair color , skin care , sun protection , make-up perfumes , and hair care .

Figure 4.2: Highlighted text shows hallucinations. (1) contains ENT hallucinations, and (2) contains ENT, and OI.

can produce better results. Kanthara et al. [2022] format the data table by following the template: *title + y-values + x-values* (see Table 4.2). This input format includes the title information but it lacks the x and y axis labels and the corresponding x-y values are not put together. The distance between two x-y values is large and we speculate that T5 and BART models must have faced some difficulty when learning the relationship between all

Input format	Platform | Facebook | 0 | bar chart 18-24 | 36 | 1 | bar chart 25-29 | 41 | 2 | bar chart 30-34 | 50 | 3 | bar chart Platform | Instagram | 0 | bar chart 18-24 | 24 | 1 | bar chart 25-29 | 17 | 2 | bar chart 30-34 | 13 | 3 | bar chart Platform | YouTube | 0 | bar chart 18-24 | 12 | 1 | bar chart 25-29 | 14 | 2 | bar chart 30-34 | 11 | 3 | bar chart Platform | LinkedIn | 0 | bar chart 18-24 | 7 | 1 | bar chart 25-29 | 10 | 2 | bar chart 30-34 | 10| 3 | bar chart Platform | Pinterest | 0 | bar chart 18-24 | 9 | 1 | bar chart 25-29 | 8 | 2 | bar chart 30-34 | 8 | 3 | bar chart Platform | Snapchat | 0 | bar chart 18-24 | 9 | 1| bar chart 25-29 | 9 | 2 | bar chart 30-34 | 5 | 3 | bar chart Platform | Twitter | 0 | bar chart 18-24 | 3 | 1 | bar chart 25-29 | 1 | 2 | bar chart 30-34 | 2 | 3 | bar chart
Gold Summary	This statistic presents the most popular social network among Millennials in the United States as of August 2016 , by age group. During the survey period , 24 percent of respondents between 18 and 24 years old stated that they used Instagram the most .
Delexicalized Gold Summary	This statistic presents the templateTitle[2] templateTitle[3] templateTitle[4] templateTitle[5] among templateTitleSubject[0] in the templateTitle[0] as of 2016 , templateTitle[7] templateTitle[8] templateTitle[9]. During the survey period , templateValue[1][1] templateScale of respondents between 18 and templateValue[1][1] years old stated that they used templateValue[0][1] the templateTitle[2]
Delexicalized Generated Summary	This statistic shows the results of a survey on the templateTitle[0] templateTitle[1] templateTitleSubject[0] templateTitle[3] in templateTitleSubject[1] as of 2018 , sorted templateTitle[7] templateTitle[8] . During the survey , templateValue[1][max] templateScale of respondents aged templateLabel[1][0] to templateLabel[1][1] years stated that they used Gmail as their primary templateTitle[3] provider . templateValue[0][1] was ranked second with templateValue[1][1] templateScale .
Generated Summary	This statistic shows the results of a survey on the U.S. Millennials popular in Millennials as of 2018 , sorted by age . During the survey , 36 % of respondents aged 18-24 to years stated that they used Gmail as their primary popular provider. Instagram was ranked second with 24%.

Table 4.1: Example from c2t-small dataset.

the y values and all the x values.

All the above mentioned observations lead to our first hypothesis:

> Reducing long distance dependencies between x and y axis values, and adding title and x and y axis labels in the linearized input data will alleviate hallucinations.

We propose a linearized input format that reduces long distance dependencies. The tem-

foreign born populations in millions | 50 | 40 | 30 | 20 | 10 | 1900 | 1925 | 1950 | 2000 | 1850 | 1875 | 1975

Table 4.2: Linearized input format used by Kanthara et al. [2022]. Example from c2t-big dataset.

plate we use is *title + x-y labels + x-y values*. So we put the x values with their corresponding y values, append the x and y axis labels before the coordinate values, and append the title before that. Table 4.3 shows how the single and multi column data is going to be linearized for both c2t-small and c2t-big datasets.

Single column	Most popular news brands in the United States as of June 2018 , by reach x-y labels news brand - Reach, x-y values The New York Times 26% , CNN 25% , FOX News 22% , The Washington Post 21% , Business Insider 20% , USA Today 19% , The Huffington Post 19% , MSN News 18% , CBS News 16% , Forbes 14%
Multi column	Sales volume of beer in Prince Edward Island (P.E.I) from FY 2012 to FY 2019 , by product type (in million liters) labels Year - Packaged - Draught values 2019 8.62 1.13 , 2018 8.65 1.1 , 2017 8.19 0.98 , 2016 8.48 0.91 , 2015 8.39 0.83 , 2014 8.47 0.74 , 2013 8.84 0.65 , 2012 8.79 0.64

Table 4.3: Proposed format: title + x-y labels + x-y values

4.2 Problems in the Training Summaries

We analyzed the same fifty examples but this time instead of analyzing the generated summaries, we analyze the training summaries, because hallucinations can occur due to source-reference divergence as previously mentioned in Section 3.3.

In Figure 4.1, we see that 13 out of 50 of the generated summaries contain outside information. This type of hallucination is extrinsic and is difficult to test, especially when we use automatic metrics, which evaluate reference-candidate pair and completely ignore the input data.

The purpose outside information serves in a summary is to make the summary more interesting. However, we cannot verify that information from the chart data or the chart itself. We also looked at summaries in c2t-big dataset and similar kind of outside information was present in that dataset because both datasets were scrapped from the same source (see Section 3.1) i.e. statista.com.

Figure 4.3 shows examples of additional information in the gold summaries. Essentially, when additional information is present in the training summaries, we are training our model to hallucinate. This leads us to our second hypothesis:

> Because of the additional information, the model learns unfaithful text in summaries and outputs them during the generation.

.

Index	Data	Gold Summary
1	title: Average retail price for white sugar in Canada 2015 to 2019 Data: 0 1 2 3 4 5 6 7 8 9 10 11 Month Dec Nov Oct Sep Aug Jul Jun May Apr Mar Feb Jan 2015 2.57 2.76 2.77 2.83 2.81 2.81 2.79 2.81 2.78 2.81 2.81 2.88 2016 2.64 2.68 2.72 2.8 2.88 2.84 2.75 2.76 2.78 2.69 2.73 2.77 2017 2.69 2.66 2.68 2.74 2.76 2.79 2.77 2.94 2.78 2.83 2.75 2.82 2018 2.56 2.55 2.67 2.61 2.7 2.67 2.58 2.71 2.71 2.58 2.69 2.7 2019 2.41 2.45 2.44 2.52 2.49 2.5 2.5 2.56 2.53 2.59 2.48 2.61	The average retail price for two kilograms of white sugar in Canada hit an all-time low of 2.41 Canadian dollars in December 2019 . This price has gradually decreased over time , from a monthly average of 2.79 dollars per two kilograms in 2015 . What is white sugar ? White sugar which we buy in stores is a refined sugar .
2	title: Women 's average age at first marriage in Italy 2018 , by region X_Axis['Month']: ['Aosta_Valley', 'Liguria', 'Emilia-Romagna', 'Tuscany', 'Sardinia', 'Friuli-Venezia_Giulia', 'Lazio', 'Trentino-South_Tyrol', 'Piedmont', 'Lombardy', 'Umbria', 'Marche', 'Veneto', 'Abruzzo', 'Molise', 'Basilicata', 'Apulia', 'Campania', 'Sicily', 'Calabria'] Y_Axis['Average', 'age']: ['34.7', '34.1', '33.9', '33.9', '33.9', '33.6', '33.5', '33.4', '33.3', '33.1', '33.0', '32.8', '32.8', '32.8', '32.2', '32.1', '31.6', '31.0', '30.8', '30.6']	In 2018 , the average age of Italian women walking down the aisle was of 32.5 years . From the perspective of the singular regions , the oldest females to tie the knot were citizens of Aosta Valley and Liguria , where the average age of the bride at the first marriage reached 34.7 years and 34.1 years in 2018 . Aosta Valley was also the region with the oldest grooms in the country – a male inhabitant of the region got married at the average age of 38.2 years .
3	title: Inflation rate in India 2024 X_Axis['Year']: ['2024', '2023', '2022', '2021', '2020', '2019', '2018', '2017', '2016', '2015', '2014', '2013', '2012', '2011', '2010', '2009', '2008', '2007', '2006', '2005', '2004', '2003', '2002', '2001', '2000', '1999', '1998', '1997', '1996', '1995', '1994', '1993', '1992', '1991', '1990', '1989', '1988', '1987', '1986', '1985', '1984'] Y_Axis['Inflation', 'rate', 'compared', 'to', 'previous', 'year']: ['3.97', '3.98', '4.05', '4.07', '4.09', '3.44', '3.43', '3.6', '4.5', '4.9', '5.8', '9.4', '10', '9.5', '10.53', '12.31', '9.09', '6.2', '6.7', '4.4', '3.82', '3.86', '3.98', '4.31', '3.83', '5.7', '13.13', '6.84', '9.43', '9.96', '10.28', '7.28', '9.86', '13.48', '11.2', '4.57', '7.21', '9.06', '8.89', '6.25', '6.52']	The statistic shows the inflation rate in India from 1984 to 2018 , with projections up until 2024 . The inflation rate is calculated using the price increase of a defined product basket . This product basket contains products and services , on which the average consumer spends money throughout the year .

Figure 4.3: Highlighted text shows hallucinations in the gold summaries.

In Section 3.3, we talked about the fact that datasets which are built using heuristics can have source-reference divergence. Like WIKIBIO, authors of c2t-big and c2t-small crawled statista website and downloaded charts, data table, axis labels, and human-written summaries. These human-written summaries contain additional information that cause the model to hallucinate.

5. Testing the Two Hypotheses

In this chapter, we prove the two hypotheses we formulated in the previous chapter. In Section 5.1 we show that formatting of input data reduces hallucinations and in Section 5.2, we show that additional information in the training summaries leads to hallucinations.

5.1 Experiments and Results for Hypothesis I

In Section 4.1, we formulated our first hypothesis:

> Reducing long distance dependencies between x and y axis values, and adding title and x and y axis labels in the linearized input data will alleviate hallucinations.

5.1.1 Experiments

We train three models using T5. First model, called *t5+c2t-small+O&H*, which was trained on linearized data as represented by Obeid and Hoque [2020] (see Table 4.1) on c2t-small. Second model, called *t5+c2t-small, our linearization*, was trained on our proposed linearized input format as shown in table 4.3. Third model, called *t5+c2t-big, our linearization*, was trained on proposed linearized input format as shown in table 4.3 for c2t-big data. The data splits are the same as mentioned in table 3.1. As a prefix, we use *'C2T: '* before every instance of input data. The training details are given in Appendix A.1.

t5+c2t-small+O&H and t5+c2t-small were evaluated on BLEU-4 using the sacreBLEU library [Post, 2018], ROUGE-2, and NUBIA. t5+c2t-big was evaluated on BLEU, ROUGE-2, BLEURT, GPT-2 perplexity[1], and NUBIA.

5.1.2 Results

Model	BL	RG-2	L	C	N	SS	N
Obeid and Hoque [2020]	18.5						
t5+c2t-small, O&H linearization	26.1	33.5	5.5	67.8	26.5	3.0/5	35.4
t5+c2t-small, our linearization	33.9	44.8	33.2	22.3	44.4	3.5/5	46.9

Table 5.1: BL is BLEU-4 score, RG-2 is ROUGE-2 score, L is Logical Agreement, C is Contradiction, N is Neutrality, SS is Semantic Similarity, and N is the final NUBIA score. We can see in these two tables that our linearization improves the results for c2t-small dataset.

We can see in table 5.1 and 5.2 that reducing long distance dependencies, and adding more chart information in the linearized input, results in higher scores across all metrics. This improvement is quite simple and intuitive. First improvement is done by using T5 on the original input format. This results in improving BLEU score by 8 points. However, the problem with using Obeid and Hoque [2020] input format is that it results in a lot of entity hallucinations and that is reflected in the second table. Logical agreement is very low and

[1] https://huggingface.co/docs/transformers/perplexity

Model	BL	RG-2	PPL	BLT	L	C	N	SS	N
Kanthara et al. [2022] T5	37.0		10.0	0.15					
t5+c2t-big, our linearization	39.8	55.0	8.2	0.17	39.3	21.3	39.3	3.6/5	55.6

Table 5.2: BL is BLEU-4 score, RG-2 is ROUGE-2 score, PPL is Perplexity, BLT is BLEURT score, L is Logical Agreement, C is Contradiction, N is Neutrality, SS is Semantic Similarity, and N is the final NUBIA score. We can see in these two tables that our linearization improves the results for c2t-big dataset. Kanthara et al. [2022] did not compute ROUGE-2 and NUBIA but the results show that by using our linearization, the contradiction score is low.

contradiction is very high. By error analysis, we came to the conclusion that the model was generating random named entities it learned through the training summaries.

If the title of the chart data is in the input, all the scores improve along with NUBIA because then the model learns to pick entities from the input instead of generating named entities from parametric knowledge. Similarly, the improvement can be seen when trained on c2t-big dataset. They did not append the *x-y* labels in the linearized input (see Table 4.2). This small addition results in improvement of 2.81 BLEU score. An unforseen advantage of the new setup was that our method produced better results in only **14 epochs**, compared to Kanthara et al. [2022], who trained for **82 epochs**.

5.1.3 Error Analysis

Now we analyze the text generated from the *t5+c2t-small, our linearization* in detail. We look at randomly picked 50 examples from the test set and analyze the hallucinations that are present in the text given the input chart data.

All the summaries that we analyzed were neither incoherent nor did they have repetitions. We think this is largely due to using a pre-trained transformer model.

Most of the generated summaries correctly picked up the discourse structure of the dataset. The generated summaries first mention what the chart is about and then it talks about one or two statistics from the chart data. There are some instances where the summary does not contain any statistic from the chart and it contains only extrinsic hallucinations (OI).

Amazon : annual gross merchandise value 2012 to 2015 x-y labels Year - GMV in billion U.S: dollars, x-y values 2012 108.0 , 2013 136.0 , 2014 166.0 , 2015 225.6
This statistic presents Amazon's gross merchandise value from 2012 to 2015. In the most recently reported fiscal period, the online retailer's gross merchandise value amounted to 225.6 billion U.S. dollars, up from 136 billion U.S. dollars in 2013. Amazon.com, an American electronic commerce company originally founded in 1994, is the world's largest online retailer of books, clothing, electronics, music and many more.

Table 5.3: Generated summary that correctly talks about values from the data table but contains extrinsic hallucination.

For example, if we look at Table 5.3, the generated text looks fluent and coherent. The

statistic '225.6 billion U.S. dollars, up from 136 billion U.S. dollars in 2013' is also correct. However, the last sentence 'Amazon.com, an American electronic commerce company originally founded in 1994, is the world's largest online retailer of books, clothing, electronics, music and many more.' is nowhere near to be found in the chart data. We understand that this statement is correct. Amazon.com is an American electronic commerce company and it was indeed founded in 1994. However, we do not get this information from the chart and therefore we cannot verify it. Looking at the chart, we do not know if Amazon.com is the largest online retailer for books, clothing, etc., and that it was founded in 1994. Looking at the chart, we can only tell what Amazon's annual gross merchandise values were from 2012 to 2015. This type of hallucination is called extrinsic hallucination (see Section 2.1.3).

Table 5.4 shows another example where this type of problem lies. The sentences 'GDP refers to the total market value of all goods and services that are produced within a country per year. It is an important indicator of the economic strength of a country.' are not in the chart data. Both of the sentences are extrinsic hallucinations.

Gross domestic product (GDP) growth rate in India 2024 x-y labels Year - GDP growth rate compared to previous year, x-y values 2024* 7.33% , 2023* 7.44% , 2022* 7.44% , 2021* 7.44% , 2020* 7.03% , 2019* 6.12% , 2018 6.81% , 2017 7.17% , 2016 8.17% , 2015 8% , 2014 7.41%
The statistic shows the growth rate of the real gross domestic product (GDP) in India from 2014 to 2018, with projections up until 2024. GDP refers to the total market value of all goods and services that are produced within a country per year. It is an important indicator of the economic strength of a country.

Table 5.4: Generated summary with only extrinsic hallucination.

5.2 Experiments and Results for Hypothesis II

In Section 4.2, we formulated our second hypothesis:

> Because of the additional information, the model learns unfaithful text in summaries and outputs them during the generation.

.

5.2.1 Experiments

We train two models using T5 on the Autochart dataset. The reason we use the Autochart dataset is that we assume this dataset does not contain any hallucinations. The assumption is based on the fact that text in the dataset were constructed using a template which only utilizes chart related information (see Section 3.2.3).

Our first model in this experiment is called *t5-Autochart-Original*, which is trained on linearized data as represented in Table 4.3 with original summaries. Second model, called *t5-Autochart-Noise*, is trained on summaries that are augmented with noise generated from GPT-2. Essentially, this is done to check if the generated summaries contain extrinsic hallucinations, like in Obeid and Hoque [2020] and Kanthara et al. [2022]. We use 8000 summaries as training, 1296 for validation, and 1297 for testing. The training details are

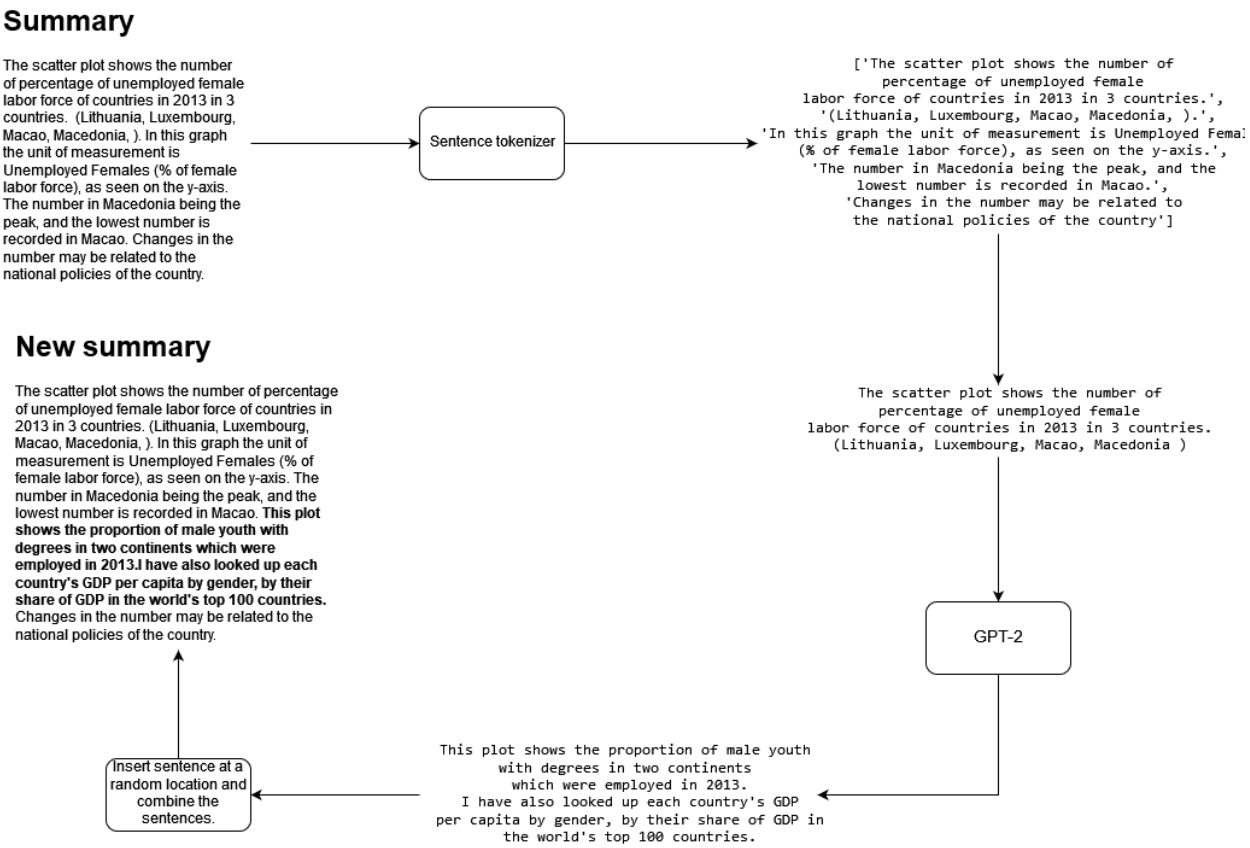


Figure 5.1: Noise generation flow. Segment each summary, pick the first two sentences and pass it as a prompt to GPT-2. Insert the generated output from GPT-2 to original summary at a random location. Text in bold show the output generated by GPT-2.

given in Appendix A.1. *t5+Autochart+** is evaluated on BLEU, ROUGE-2, BLEURT, and NUBIA.

To inject noise in the summaries, we first segment the summary using NLTK [Bird et al., 2009] sentence tokenizer. After segmenting the summary, we randomly pick a sentence and give it as the input to the GPT-2 model. For GPT-2 generation, we use greedy search. The generated sentence is then inserted at a random location in the tokenized sentence list, and then all the sentences are combined. The noise generation flow is shown in Figure 5.1, and Table 5.5 show the original and the noisy summary. As a prefix, we use *'C2T: '* before every instance of input data.

5.2.2 Results

Results of our second experiment are reported in table 5.6. The BLEU, ROUGE, and BLEURT scores confirm our hypothesis that hallucinations occur due to noise in the training summaries. However, the NUBIA score gives mixed results. On one hand, the logical agreement for original data is more than logical agreement of noisy data but on the other hand, noisy data has lower contradiction and higher semantic similarity score. However, if we look at the neutrality score, it is much higher and it seems that NUBIA is not contradicting

Original Summary	The scatter plot shows the number of percentage of unemployed female labor force of countries in 2013 in 3 countries. (Lithuania, Luxembourg, Macao, Macedonia,). In this graph the unit of measurement is Unemployed Females (% of female labor force), as seen on the y-axis. The number in Macedonia being the peak, and the lowest number is recorded in Macao. Changes in the number may be related to the national policies of the country.
Noisy Summary	The scatter plot shows the number of percentage of unemployed female labor force of countries in 2013 in 3 countries. (Lithuania, Luxembourg, Macao, Macedonia,). In this graph the unit of measurement is Unemployed Females (% of female labor force), as seen on the y-axis. The number in Macedonia being the peak, and the lowest number is recorded in Macao. **This plot shows the proportion of male youth with degrees in two continents which were employed in 2013.I have also looked up each country's GDP per capita by gender, by their share of GDP in the world's top 100 countries.** Changes in the number may be related to the national policies of the country

Table 5.5: Original and Noisy Summary. Text in bold shows sentence generated by GPT-2.

the noisy text, thus indicating a high number of extrinsic hallucinations.
More research needs to be done on which metrics would be better for checking faithfulness. NUBIA gave us an absolute result that agreed with all the other automatic metrics for the first experiment (see Table 5.1) but for the second experiment it gave us contradictory results. It could be argued that overall NUBIA score is affected by high neutrality and low contradiction, and the only part that is relevant, is logical agreement because we are checking faithfulness and not irrelevancy or neutrality.

Model	**BL**	**RG-2**	**BLT**	**L**	**C**	**N**	**SS**	**N**
t5-Autochart-Original	58.8	62.0	0.18	33.0	31.0	35.5	3.2/5	79.8
t5-Autochart-Noise	46.0	48.9	-0.10	24.0	24.4	51.5	3.3/5	88.4

Table 5.6: BL is BLEU-4 score, RG-2 is ROUGE-2 score, BLT is BLEURT score, L is Logical Agreement, C is Contradiction, N is Neutrality, SS is Semantic Similarity, and N is the final NUBIA score.

5.2.3 Manual Analysis

We manually analyzed 50 samples from both models. Summaries generated from t5 - Autochart-Original were coherent, fluent, and did not contain any extrinsic hallucinations. However, summaries generated from t5-Autochart-Noise were repetitive, and contained extrinsic hallucinations.

In Example 5.7, we can observe how the summary generated from t5-autochart-noise is repeated twice. We speculate that repetitions occurred when the model failed to devise any un-grounded text, and ended up repeating itself. In Example 5.8, we can observe that the first line is not grounded in the data. We do not know if the data is from the UNDP website. Out of the 50 summaries analyzed, 27 contained extrinsic hallucinations, and 43 contained repetitions.

Data	Percentage of total land area in Bahrain where the elevation is below 5 meters bar_chart x-y labels Year - Elevated land area (as % of total land area), x-y values 1990 39.03492 , 2000 39.03492
t5-Autochart-Original output	This barchart shows the number of percentage of total land area in bahrain where the elevation is below 5 meters from 1990 through 2000. The unit of measurement in this graph is Elevated land area (as % of total land area) as shown on the y-axis. It is clear that the number remain constant The number in 1990 being the peak, and the lowest number is found in 1990.
t5-Autochart-Noise output	The bar graph illustrates the number of percentage of total land area in bahrain where the elevation is below 5 meters from 1990 to 2000. The unit of measurement in this graph is Elevated land area (as % of total land area) as shown on the y-axis. There is a steady trend of the number. The number in 1990 being the peak, and the lowest number is found in 1990. The bar graph illustrates the number of percentage of total land area in bahrain where the elevation is below 5 meters from 1990 to 2000. The bar graph illustrates the number of percentage of total land area in bahrain where the elevation is below 5 meters from 1990.

Table 5.7: An example of generated summaries from t5-Autochart-Orignal and t5-Autochart-Noise. Red highlights repetitions.

Data	Percentage of population having access to electricity and non-solid fuels in Serbia bar_chart Year-Access to resources (% of population)(Non-solid fuel,Electricity) x-y1-y2 values 1990 49.1546094417572 , 2000 60.6516391038895 , 2010 67.3970907926559 2012 68.63474547863011
t5-Autochart-Original output	This bar plot is about the number of percentage of population having access to electricity and non-solid fuels in serbia from 1990 until 2012. The unit of measurement in this graph is Access to resources (% of population) as shown on the y-axis. The first group data denotes the change of Non-solid fuel. It is clear that there has been an rise in the last 3 years. The highest number is found in 2012 and the lowest number is recorded in 1990. Number changes may be linked to national policies of the country. The second group data denotes the change of Electricity. Overall, the numbers of vists seem to be relatively stable The number in 1990 being the peak, and the lowest number is recorded in 1990.
t5-Autochart-Noise output	Data is from the UNDP (http://www.un.org/en/publications/data-in-serbia) (Table S1). The percentage of people accessing electrical and solid fuels is a proxy for The scatter graph illustrates the number of percentage of population having access to electricity and non-solid fuels in serbia from 1990 to 2012. In this graph the unit of measurement is Access to resources (% of population), as seen on the y-axis. The first group data denotes the change of Non-solid fuel. The rise in the number has been seen slowly, from about 49 Access to resources (% of population) in 1990 to about 68 Access to resources (% of population) in 2012. The number in 2012 being the peak, and the lowest number is found in 1990. Changes in the number may be related to the national policies of the country The second group data denotes the change of Electricity. Overall, the numbers of vists seem to be relatively stable The peak of the number is recorded in 1990 and the lowest number is found in 1990.

Table 5.8: An example of generated summaries from t5-Autochart-Orignal and t5-Autochart-Noise. Red highlights extrinsic hallucination.

6. Further Improving Faithfulness of Summaries

We saw in Section 5.2 that if the training data contains additional information, the model generates repititive and hallucinated text. Using T5 and reducing long-term dependencies (see Section 5.1) can only help to an extent. To keep the model faithful to the input data, we require training data that does not contain additional information. In this chapter, we propose a method to improve faithfulness of the summaries. Our proposed method has four steps: (1) cleaning the dataset using NLI (Section 6.1), (2) fine-tuning T5 on filtered dataset (Section 6.2), (3) create a small dataset of faithful summaries (Section 6.3), (4) introduce a new task called 'improve summary', fine-tune the model in step (2) on 'improve summary' task using faithful summaries, and introduce 2-step generation (Section 6.4).

We also conduct ablation studies to study our 2-step generation setup in Section 6.5. Lastly, we conduct human evaluation in Section 6.6 where we compare models trained using our proposed method with the baseline. The model we use as baseline in this chapter is *t5+c2t-small, our linearization*, which we trained earlier in Section 5.1.

6.1 Step 1: Cleaning the dataset using NLI

The first step to getting rid of extrinsic hallucinations is to remove all the sentences from the training summaries which are not grounded in the input data.

Recently, NLI has been used in the field of NLG as a tool to build faithful datasets [Pang et al., 2021], and for evaluating semantic accuracy [Dušek and Kasner, 2020]. We take inspiration from Dušek and Kasner [2020], and instead of using NLI as a semantic accuracy metric, we utilize it as a pre-processing tool. The idea is, if sentences in a summary are not entailed in the linearized data, we discard them.

We employ BART-NLI model[1] for dataset cleaning. BART-NLI model is a pre-trained NLI model for zero-shot sequence classification. It is trained on MultiNLI dataset [Williams et al., 2018]. Given a premise and hypothesis, the NLI model assigns a score between 1 and 100. If the hypothesis is entailed in the premise then the score is close to a hundred. This model can be used in two settings, single true class where the scores of all the hypotheses add to 100, or multiple true class, where the scores of individual hypotheses are computed separately out of 100.

To utilize NLI as a filtering step, we first segment each summary into sentences and pass those sentences as hypotheses with the corresponding linearized data as a premise, to the NLI model. We use the multiple true class setting because more than one sentence can be entailed by the data. We use a threshold of 30 to separate entailed and non-entailed sentences. If the sentence has a score of 30 or above, we keep that sentence in our summary, otherwise, we drop it. Figure 6.1 shows a diagram of the overall cleaning process. We apply the cleaning step on the entire c2t-small dataset (see Section 3.2.1), which contains 8147 examples.

To determine the threshold, we experimented with the BART-NLI model on several chart data-summary pairs. We analyzed a random sample of hundred filtered summaries from the training set of c2t-small dataset, and found out that the average score given to the entailed

[1]BART-NLI: https://huggingface.co/facebook/bart-large-mnli

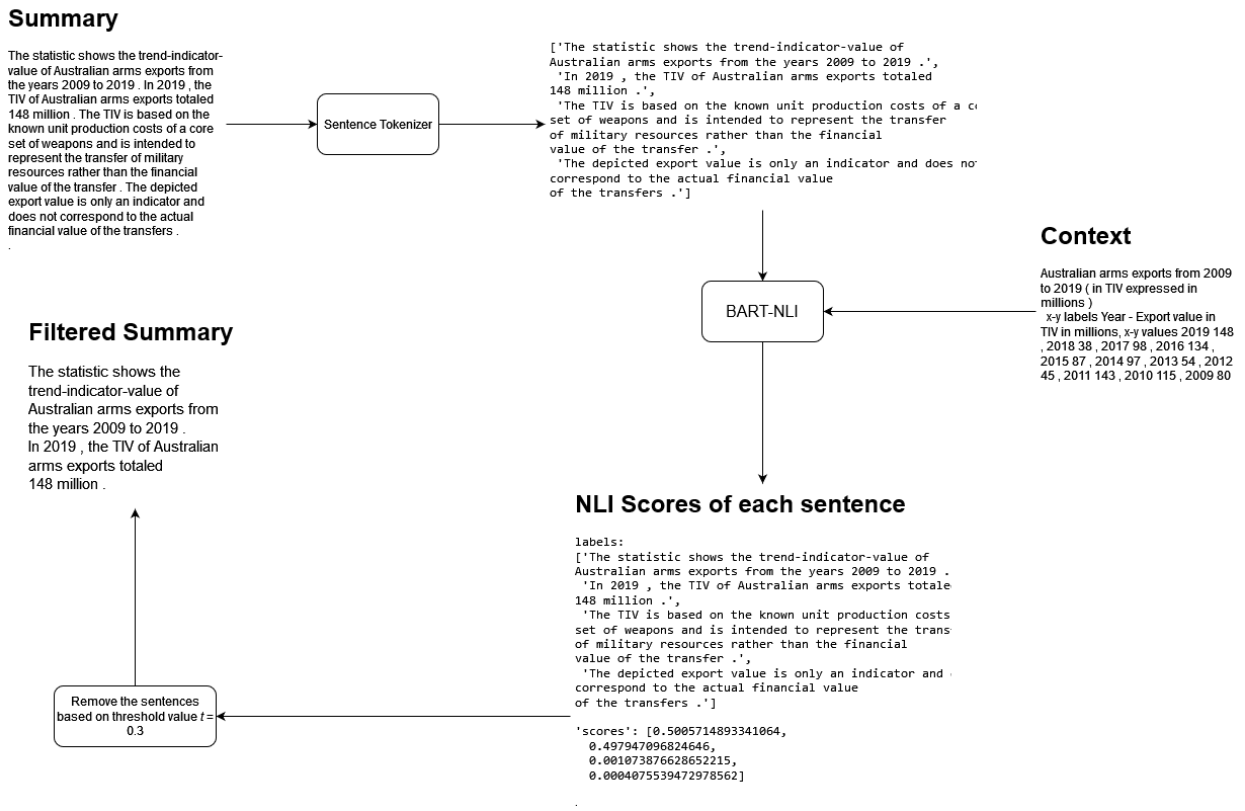


Figure 6.1: Summary cleaning process using zero-shot BART-NLI.

sentences were 89 and the average score given to the non-entailed sentences was 8.7. This means that the model is sure when assigning the score, and making minor adjustments would not lead to significant improvements. We also analyzed those hundred summaries to check if they were correctly filtered or not. Out of the 100, 69 summaries were correct and 31 summaries had a sentence which was incorrectly scored.

Data	Most common male names in Denmark 2020 x-y labels Month - Number of individuals, x-y values Peter 48011 , Jens 45000 , Michael 44811 , Lars 44370 , Thomas 41987 , Henrik 41896 , Søren 40152 , Christian 37694 , Jan 37581 , Martin 37132 , Niels 34790 , Anders 33920 , Morten 33877 , Jesper 33706 , Mads 31857 , Hans 31084 , Jørgen 31036 , Per 30636 , Rasmus 30363 , Ole 30082
Gold Summary	As of January 2019 , there were approximately 2.89 million men living in Denmark . Among these , 48 thousand men had the name Peter . It is also found in the variants Petar , Peder , Per and others .
Summary after applying NLI	Among these , 48 thousand men had the name Peter . It is also found in the variants Petar , Peder , Per and others .

Table 6.1: A good example of filtering. Summary before and after applying NLI. Red color highlights the sentence that is correctly filtered out.

We show three examples of how summaries look before and after cleaning. In Table 6.1,

Data	Gross domestic product (GDP) per capita in selected global regions 2018 x-y labels Region - GDP per capita in U.S. dollars, x-y values Africa Sub-Sahara 1585.77 , South Asia 1902.85 , Arab World 6608.81 , Latin America and Caribbean 9044.22 , East Asia and Pacific 11142.59 , Europe and Central Asia 25107.27 , EU 36569.73 , Euro area 39927.62 , North America 61117.05
Gold Summary	This statistic shows the gross domestic product (GDP) per capita in selected world regions in 2018 . In North America , the gross domestic product per capita in 2018 amounted to approximately 61,117.05 U.S. dollars .
Summary after applying NLI	This statistic shows the gross domestic product (GDP) per capita in selected world regions in 2018 .

Table 6.2: A bad example of filtering. Summary before and after applying NLI. Red color highlights the sentence that is incorrectly filtered out.

we see that the NLI perfectly filters the summary by removing the sentence about 'total population of men in Denmark'. There is no information about the total population present in the data. In the Table 6.2, we can see that the BART-NLI model decides that the sentence is not entailed in the data even if the sentence talks about some statistics from the data. The sentence states the value as '61,117.05' and the data states the value as '61117.05'. We speculate that the BART-NLI model thinks that these two values are different because of a comma - hence, gives it a low score. Lastly, we can see in the example in Table 6.3 that no filtering was done. The correct filtering for this summary would lead to only the first sentence 'The statistic shows gross domestic product (GDP) per capita in Kenya from 1984 to 2024.'. However, the BART-NLI model inferred that all the sentences are contained in the summary.

Data	Gross domestic product (GDP) per capita in selected global regions 2018 x-y labels Region - GDP per capita in U.S. dollars, x-y values Africa Sub-Sahara 1585.77 , South Asia 1902.85 , Arab World 6608.81 , Latin America and Caribbean 9044.22 , East Asia and Pacific 11142.59 , Europe and Central Asia 25107.27 , EU 36569.73 , Euro area 39927.62 , North America 61117.05
Gold Summary	The statistic shows gross domestic product (GDP) per capita in Kenya from 1984 to 2024. GDP is the total value of all goods and services produced in a country in a year. It is considered to be a very important indicator of the economic strength of a country and a positive change is an indicator of economic growth.
Summary after applying NLI	The statistic shows gross domestic product (GDP) per capita in Kenya from 1984 to 2024 . GDP is the total value of all goods and services produced in a country in a year. It is considered to be a very important indicator of the economic strength of a country and a positive change is an indicator of economic growth.

Table 6.3: A bad example of NLI cleaning. Text shown in red should not be in the summary. However, BART-NLI determines that the text is contained in the data.

Overall, we think that the results produced by the BART-NLI model are fairly decent

considering we are not fine-tuning it on our particular task, which is to check whether the sentences are entailed in the linearized data or not. Secondly, we want to remove as much additional information in the summaries as possible, as quickly as possible, and using BART-NLI does that job.

6.2 Step 2: Fine-tuning T5 on filtered dataset

The next step is to fine-tune a T5 model on the filtered c2t-small dataset. This is the exact same dataset as the baseline with cleaning applied and the dataset splits are the same, as mentioned in Table 3.1. The task prompt is *'C2T: '*. We call this model *NLI+T5*.

Results and Error Analysis

Table 6.4 shows the results of the T5 model fine-tuned on the filtered data. We analyzed the same 50 data points that we analyzed previously in Section 5.1.2, to see what changes were made to the summaries.

As previously mentioned, the main idea of using the NLI model as a pre-processor was to get rid of sentences that were not grounded in the input chart data. The NLI model mostly removed the sentences that were there as additional information. And, as we previously saw in experiments for our Hypothesis II in Section 5.2, additional information causes extrinsic hallucinations. We manually analyzed 50 summaries and out of those 50 summaries, 46 summaries did not contain any hallucinations.

Model	**BL**	**RG-2**	**BLT**	**PPL**	**L**	**C**	**N**	**SS**	**N**
Baseline	33.9	44.8	0.087	7.5	33.2	22.3	44.4	3.5/5	46.9
NLI+T5	34.2	43.7	0.018	7.1	33.1	10.2	56.5	3.5/5	44.5

Table 6.4: BL is BLEU-4 score, RG-2 is ROUGE-2 score, BLT is BLEURT score, PPL is Perplexity, L is Logical Agreement, C is Contradiction, N is Neutrality, SS is Semantic Similarity, and N is the final NUBIA score.

The result of training a model on filtered data is that it often produces summaries of shorter length (just one sentence). NLI+T5 generated 28% of the summaries of just a single sentence. Compared to the baseline, there were zero summaries of length one. We expected this problem because our method of filtering only removes the sentences. It does not replace the sentence(s) with a sentence(s) that is entailed in the data.

In the example in Table 6.5, we can see the two summaries generated by the baseline and NLI+T5 respectively. Both summaries are equally faithful but the baseline summary is better because it is more informative.

Note that NLI+T5 produces single sentence summaries for all the input data. 65% of the summaries are of length two. In the example in Table 6.6, we see how the generated summary is of adequate length and is one 100% faithful to the input data.

We see that there are two problems after applying NLI, (1) summaries are not always filtered correctly, and (2) some of the summaries only contain one sentence which leads to the model generating a single sentence summaries.

Data	Time spent on reading the Bible in the U.S. from 2013 to 2017 labels Year - 2013 - 2014 - 2015 - 2016 - 2017 values 1 hour or more 24% 21% 27% 27% 23% , 45-59 minutes 8% 7% 6% 4% 6% , 30-44 minutes 26% 30% 30% 30% 29% , 15-29 minutes 27% 26% 24% 26% 25% , Less than 15 minutes 15% 15% 13% 13% 17%
Baseline T5 output	This survey shows the results of a survey regarding the question Approximately how much time do you spend on reading the Bible? " from 2013 to 2017. During the most recent survey, 23 percent of respondents stated that they spend one hour or more on reading the Bible.
NLI+T5 output	This statistic shows the average amount of time spent reading the Bible in the United States from 2013 to 2017.

Table 6.5: Difference of using filtered summaries and original summaries. NLI+T5 produces a faithful summary of one sentence.

Data	Leading car model registered in Italy 2019 x-y labels Car Model - Number of registrations, x-y values Fiat Panda 138132 , Lancia Ypsilon 58759 , Dacia Duster 43701 , Fiat 500X 42554 , Renault Clio 41792 , Jeep Renegade 41683 , Citroen C3 41646 , Volkswagen T-Roc 39600 , Toyota Yaris 36805 , Jeep Compass 35568 , Fiat 500 35493 , Dacia Sandero 35340 , Volkswagen Polo 34629 , Renault Captur 32184 , Ford Fiesta 31791
Baseline T5	In 2019, the Fiat Panda was the most registered passenger car model in Italy. With more than 138 thousand registrations, the Italian car brand was followed by Lancia Ypsilon, with over 58 thousand registrations. The most registered passenger car models were SEAT Leon and SEAT Ibiza, both exceeding 30 thousand registrations during 2018.
NLI+T5	In 2019, Fiat Panda was the leading car model registered in Italy, with over 138 thousand registrations. The second largest car manufacturer was Lancia Ypsilon with over 58 thousand registrations.

Table 6.6: Difference of using filtered data and original data. NLI+T5 produces a summary without extrinsic hallucinations.

6.3 Step 3: Dataset for further few-shot training

As we saw earlier, the summaries filtered out by the BART-NLI model can still contain sentences that are not grounded in the chart data. Secondly, the problem with the above-generated summaries is that they are too short. 28% of the times, it is just a single sentence summary.

To remedy the above two problems, we create a small training and validation set out of 330 examples from the 'filtered' validation set. We manually check each summary if it contains a sentence with additional information. If so, we remove the additional information sentence and replace it with a new manually written sentence grounded in the chart. We also check if the summary is only a single sentence, and if so, we add a manually written

sentence that is grounded in the chart. This way, we get 280 training and 50 validation examples of summaries that do not contain any hallucinations. We further annotated 150 examples from the 'filtered' test set and repeat the above process again to get a small test set.

6.4 Step 4: Further fine-tuning of the T5 model

In the last step, we aim to do 2-step generation. The first step of the generation is to generate summary using 'C2T:' task. The second step generation is to improve the summary. To this purpose, we perform a new task called *'improve summary'* using our manually annotated dataset. In this new task, we modify the input. In all our previous experiments, the input only contained linearized input data. But for this task, we append the filtered summary from the validation set to the linearized input data. An example of this format is shown in Table 6.7. The idea is to improve the summary by either removing any additional information or generate additional sentences for the summaries or keep the summary (if it is of adequate length and does not contain additional information).

|**Data**| Detroit Tigers all-time home run leaders 2019 x-y labels Players - Number of home runs, x-y values Al Kaline 399 , Norm Cash 373 , Miguel Cabrera 339 , Hank Greenberg 306 , Willie Horton 262 , Cecil Fielder 245 , Lou Whitaker 244 , Rudy York 239 , Lance Parrish 212 , Bill Freehan 200. |**summary**| This statistic shows the Detroit Tigers all-time home run leaders as of October 11, 2019.

Table 6.7: Modified input format for few-shot fine-tuning.

We re-train the T5 model trained in Step 2 on the 'improve summary' task. We use the same hyper-parameters as before and fine-tune the model for 6 epochs. The prefix we use is 'improve Summary: '. We call this model, *NLI+T5+280*. The idea is that we pre-train (in a weak sense) the model on the full (filtered) dataset (NLI+T5) and then do few-shot fine-tuning to improve the summaries. Figure 6.2 shows how the inference is done for this setup.

Results and Manual Error Analysis

Table 6.8 shows the results of fine-tuning the model on 280 samples. The test set size for all three models in Table 6.8 is 150.

Model	**BL**	**RG-2**	**BLT**	**PPL**	**L**	**C**	**N**	**SS**	**N**
Baseline	32.5	43.9	0.05	7.4	30.8	23.3	45.8	3.5/5	44.0
NLI+T5	35.0	44.1	0.02	7.1	36.8	8.9	54.1	3.6/5	47.1
NLI+T5+280	65.5	69.1	0.27	11.6	46.2	5.8	47.8	3.9/5	63.7

Table 6.8: BL is BLEU-4 score, RG-2 is ROUGE-2 score, BLT is BLEURT score, PPL is Perplexity, L is Logical Agreement, C is Contradiction, N is Neutrality, SS is Semantic Similarity, and N is the final NUBIA score.

In terms of faithfulness, out of the fifty summaries, only two summaries had extrinsic hallucinations, one of which is shown in Table 6.9. The perplexity score in Table 6.8

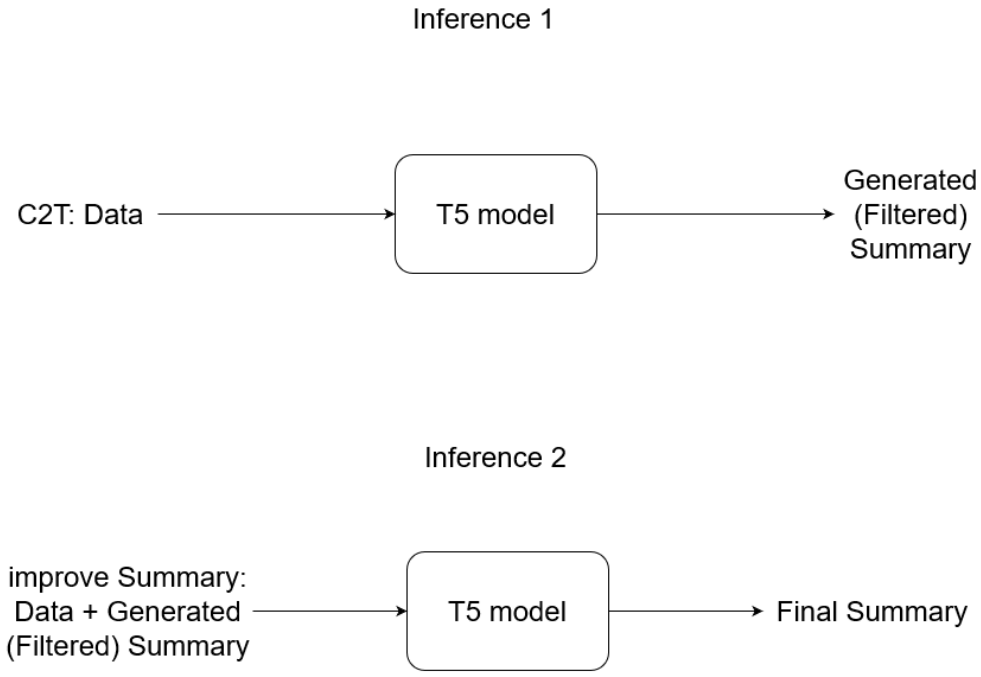


Figure 6.2: Two step inference. First, a summary is generated using the 'C2T' prefix then the summary is improved using the 'improve Summary' prefix.

of NLI+T5+280 model is relatively higher than the perplexity scores of the baseline and NLI+T5, and that is because the model suffers from repetitions. Table 6.10 shows a summary where the first sentence is repeated at the end of the summary.

Data	Gross domestic product (GDP) per capita in Kenya 2024 x-y labels Year - GDP per capita in U.S. dollars, x-y values 2024 2814.87 , 2023 2632.6 , 2022 2454.6 , 2021 2294.24 , 2020 2151.5 , 2019 1997.55 , 2018 1830.59 , 2017 1683.95 , 2016 1522.29 , 2015 1453.3 , 2014 1431.32 , 2013 1318.79 , 2012 1238.88 , 2011 1054.98 , 2010 1038.95 , 2009 982.0 , 2008 978.45 , 2007 895.23 , 2006 743.44 , 2005 621.28 , 2004 549.16 , 2003 524.74 , 2002 474.0 , 2001 479.6 , 2000 479.3 , 1999 500.11 , 1998 564.74 , 1997 507.78 , 1996 516.17 , 1995 468.0 , 1994 380.18 , 1993 326.99 , 1992 484.7 , 1991 506.8 , 1990 552.68 , 1989 546.95 , 1988 570.52 , 1987 569.04 , 1986 536.79 , 1985 467.4 , 1984 485.69
Generated Summary	The statistic shows gross domestic product (GDP) per capita in Kenya from 1984 to 2024. GDP is the total value of all goods and services produced in a country in a year. In 2018, it was around 1.83 trillion U.S. dollars and a positive change is an indicator of economic growth.

Table 6.9: An example with extrinsic hallucinations.

In terms of the length of the summaries, the problem is not one hundred percent eliminated. There were two single-sentence summaries and the rest of the summaries were of either length two or three.

Data	Most common male names in Denmark 2020 x-y labels Month - Number of individuals, x-y values Peter 48011 , Jens 45000 , Michael 44811 , Lars 44370 , Thomas 41987 , Henrik 41896 , Søren 40152 , Christian 37694 , Jan 37581 , Martin 37132 , Niels 34790 , Anders 33920 , Morten 33877 , Jesper 33706 , Mads 31857 , Hans 31084 , Jørgen 31036 , Per 30636 , Rasmus 30363 , Ole 30082
Generated Summary	As of January 2020, the most common male name in Denmark was Peter, with over 40.000 individuals having the name. As of that time, Jens was ranked second with over 45,000 individuals having the name. As of January 2020, the most common male name in Denmark was Peter, with over 40.000 individuals having the name.

Table 6.10: An example with repetition.

6.5 Ablation Studies

We conduct two ablation studies to distinguish the effects of 2-step generation and of the manually annotated dataset. The two models trained in these studies are compared to NLI+T5+280 (see Section 6.4). Both the models are tested on the same samples NLI+T5+280 was tested on.

6.5.1 Effect of 2-Step Generation

To study the effect of 2-step generation, we train the NLI+T5 model on manually annotated dataset and do 1-step generation where the task is to only generate the summary, not to improve it. We want to determine if using the manually annotated dataset after the filtered NLI, improves the summaries in terms of faithfulness and length, without a different task prompt. If the generated summaries are better from our 2-step generation, the separate 'C2T' and 'improve Summary' tasks will be rendered useless. The hyper-parameters we use are the same hyper-parameters we used in Section 6.2. We use 'C2T: ' as a prefix, and call the model trained for this study *NLI+T5+280 1-step*.

Model	**BL**	**RG-2**	**BLT**	**PPL**	**L**	**C**	**N**	**SS**	**N**
NLI+T5+280	65.5	69.1	0.27	11.6	46.2	5.8	47.8	3.9/5	63.7
NLI+T5+280 1-step	66.4	69.8	0.29	11.7	37.3	12.8	49.7	3.8/5	56.7
NLI+T5 2-step	61.8	67.2	0.20	11.4	44.2	5.9	49.8	3.9/5	63.3

Table 6.11: BL is BLEU-4 score, RG-2 is ROUGE-2 score, BLT is BLEURT score, PPL is Perplexity, L is Logical Agreement, C is Contradiction, N is Neutrality, SS is Semantic Similarity, and N is the final NUBIA score.

Looking at the results of automatic metrics in Table 6.11, we can note that BLEU, ROUGE-2, and BLEURT are slightly better than NLI+T5+280, and perplexity, NUBIA, and its feature scores are worse. Analyzing the same 50 summaries analyzed in error analysis for Step 4 (see Section 6.4), we found that 3/50 summaries were of length one. In terms of faithfulness, 45/50 summaries contained no hallucinations, which is three more than the

summaries generated from NLI+T5+280. Out of the five hallucinating summaries, four had extrinsic hallucinations, and one had intrinsic hallucinations which is shown in Table 6.12. So far we have not come across any intrinsic hallucination during the manual analysis of our systems. Another important aspect to note in Table 6.11 is the perplexity scores of NLI+T5+280 and NLI+T5+280 1-step, which are close, meaning both models produce a similar amount of repetitions. 1-step and 2-step generation yield similar results with the problems like repetitions and extrinsic hallucination in commonality. We are inconclusive as to why the model generated a summary with an intrinsic hallucination.

Data	Worlds' most dangerous cities, by murder rate 2018 x-y labels City - Murder rate per 100,000 inhabitants, x-y values Tijuana - Mexico 138.26 , Acapulco - Mexico 110.5 , Caracas - Venezuela 99.98 , Ciudad Victoria - Mexico 86.01 , Ciudad Juarez - Mexico 85.56 , Irapuato - Mexico 81.44 , Ciudad Guayana - Venezuela 78.3 , Natal - Brazil 74.67 , Fortaleza - Brazil 69.15 , Ciudad Bolivar
Generated Summary	This graph shows the worlds's most dangerous cities, by murder rate in 2018. According to the source, Tijuana was the safest city in the world with a murder rate of 138.26 murders per 100,000 inhabitants.

Table 6.12: Text in red highlights instrinsic hallucinations.

6.5.2 Effect of Manually Annotated Dataset

To study the effect of our manually annotated dataset, we pick 280+50 samples for training from the validation set and 150 samples from the test set, of filtered c2t-small dataset obtained in Step 2 (Section 6.2), which contain at least two sentences in the summaries and train a model called *NLI+T5 2-step* to do 2-step generation. We train the model for the second step ('improve summary') based on the following modification of the 280+50 samples; we remove one sentence at random from the summary for the input (see Table 6.7), but retain the full summary in the target, effectively training the model to add the artificially removed sentence back into the summary.

We want to see here if our manually annotated data provides any advantage in terms of faithfulness and length, or if it is better to pick the summaries from the filtered NLI dataset and do not perform any manual annotation.

Table 6.11 shows the results of this experiment. We can see that our manually annotated summaries provide a slight advantage. During error analysis, we found that, in terms of faithfulness, 6/50 summaries contained extrinsic hallucinations, which is four more than summaries generated from NLI+T5+280.

In conclusion, 2-step generation produces slightly better results in terms of hallucinations and length, but the effect of performing 2-step generation is not significant. And our manually annotated dataset improves the generated summaries in terms of faithfulness because the training summaries do not contain any un-grounded information. The improvement in our system comes mainly from the manually annotated dataset.

6.6 Human Evaluation

Measured Properties

We conduct human evaluation to evaluate the three main systems that we have built in our work. First system is the baseline trained in Section 5.1. Second system is NLI+T5 trained in Section 6.2, and the third system is NLI+T5+280 trained in Section 6.4. Properties we measure in the evaluation are as follows:

- **Value Correctness (VC)**: Numbers/figures/values in the summary are from the chart. The annotator determines which of the summaries are accurate. Value correctness property is a measure of factual correctness.
- **Outside Information Presence (OIP)**: Information that is not from the chart at all. This property measures extrinsic hallucinations.
- **Informativeness (INFO)**: The summary conveys a lot of information about the chart.
- **Coherence (CO)**: The information included in the summary is orderly and logically consistent.
- **Fluency (FLUE)**: Summary is grammatically correct and does not contain any repetitions.

We break factual correctness, a measure for faithfulness into two measures i.e. value correctness and outside information presence. One measures faithfulness related to values in the chart data and the latter measures extrinsic hallucinations that are produced as a result of ungrounded information in the training summaries (see Section 5.2). We include informativeness because we also want to measure which system conveys the most amount of information from the chart. Some summaries can be long, yet they do not contain any information related to the chart data, and some summaries are short yet they can contain more information from the chart data. So, measuring informativeness will help us understand which model produces the most informative text.

VC gives us a binary scores, meaning, either the summary has correct values or not. Similarly for OIP, we also get binary scores. For, INFO, CO, and FLUE, we get scores out of 5-point Likert scale [Likert, 1932], 5 being the highest score, and 1 being the lowest score.

Evaluation Setup

We conduct the human evaluation survey on 50 samples of each model. We divided these 50 samples in to 10 samples per experiment. Each experiment was annotated by 5 users. In total, 25 annotators were employed in this survey.

Results

The averages of each property (value correctness, outside information presence, informativeness, coherence, and fluency) are shown in Table 6.13.

To see if there is any significant difference between the results of three models, we conduct two types of statistical significance test. The first one is the chi-squared test [Pearson, 1900]

Model	VC ↑	OIP ↓	INFO ↑	CO ↑	FLUE ↑
Baseline T5	56.00%	38.00%	3.80/5	3.81/5	3.88/5
NLI+T5	76.00%	17.00%	3.60/5	3.91/5	3.96/5
NLI+T5+280	66.00%	20.80%	3.71/5	3.91/5	3.85/5
X^2-value	22.28	33.65	-	-	-
ANOVA F - statistic	-	-	0.08738	1.10739	0.77012
p-value < 0.05	0.000015	0.00001	0.9136	0.333218	0.4648

Table 6.13: VC is Value Correctness, OIP is Outside Information Presence, INFO is Informativeness, CO is Coherence, and FLUE is fluency. NLI+T5 model is the model obtained from training on filtered summaries using BART-NLI model (Section 6.2). NLI+T5+280 is the model that is further fine-tuned on 280 manually annotated summaries (Section 6.4).

performed on value correctness (VC) and outside information presence (OIP). The second test we perform is one-way ANOVA [Wilkinson, 1999], on informativeness (INFO), coherence (CO), and fluency (FLUE). The critical value p for both tests is 0.05. The null hypothesis H_0 for both tests is that 'there is no significant difference between the three models.' We perform chi-squared test on VC and OIP, because, the answer for each instance is either yes or no (binary). Chi-squared test is performed for each property for each model. One-way ANOVA test is performed on INFO, CO, and FLUE because the scores for these criteria are interval scale scores. We compute ANOVA F-measure on per instance and per participant levels.

We see from Table 6.13 that the three models differ significantly in terms of VC and OIP, but not in terms of INFO, CO, and FLUE. This makes sense considering the difference between the average ratings of INFO, CO, and FLUE is not large.

From the results, we can observe that NLI+T5+280 offers a compromise between VC and OIP, and INFO. In theory, we expected NLI+T5+280 to have the best scores because the 'improve' summary task should make the summaries have high VC and low OIP. However, it is understandable that NLI+T5 has the best scores, in terms of VC and OIP, because it was only trained on filtered data.

We speculate that NLI+T5+280 had low scores, compared to NLI+T5, due to the small size of manually annotated dataset. It could be argued that when the model tried to fix the length from the first inference, it could not devise any un-grounded text, and produced a text from the parametric knowledge instead of the input.

Compensation

All the annotators were from the United Kingdom and each annotator was paid according to the hourly minimum wage in the United Kingdom. The hourly minimum wage in the United Kingdom is 9.5 GBP. The annotators were paid immediately after the results were analyzed.

Difficulty for the Evaluators

During our analysis, we observed that the human evaluators had difficult time understanding charts. Some of the charts seemed confusing for the evaluators and that resulted in incorrect evaluation. For example, Figure 6.3 shows one such case where the value '87.2' is not convenient to observe. One annotator rushed to an answer and said that the values in the

summary are not correct.

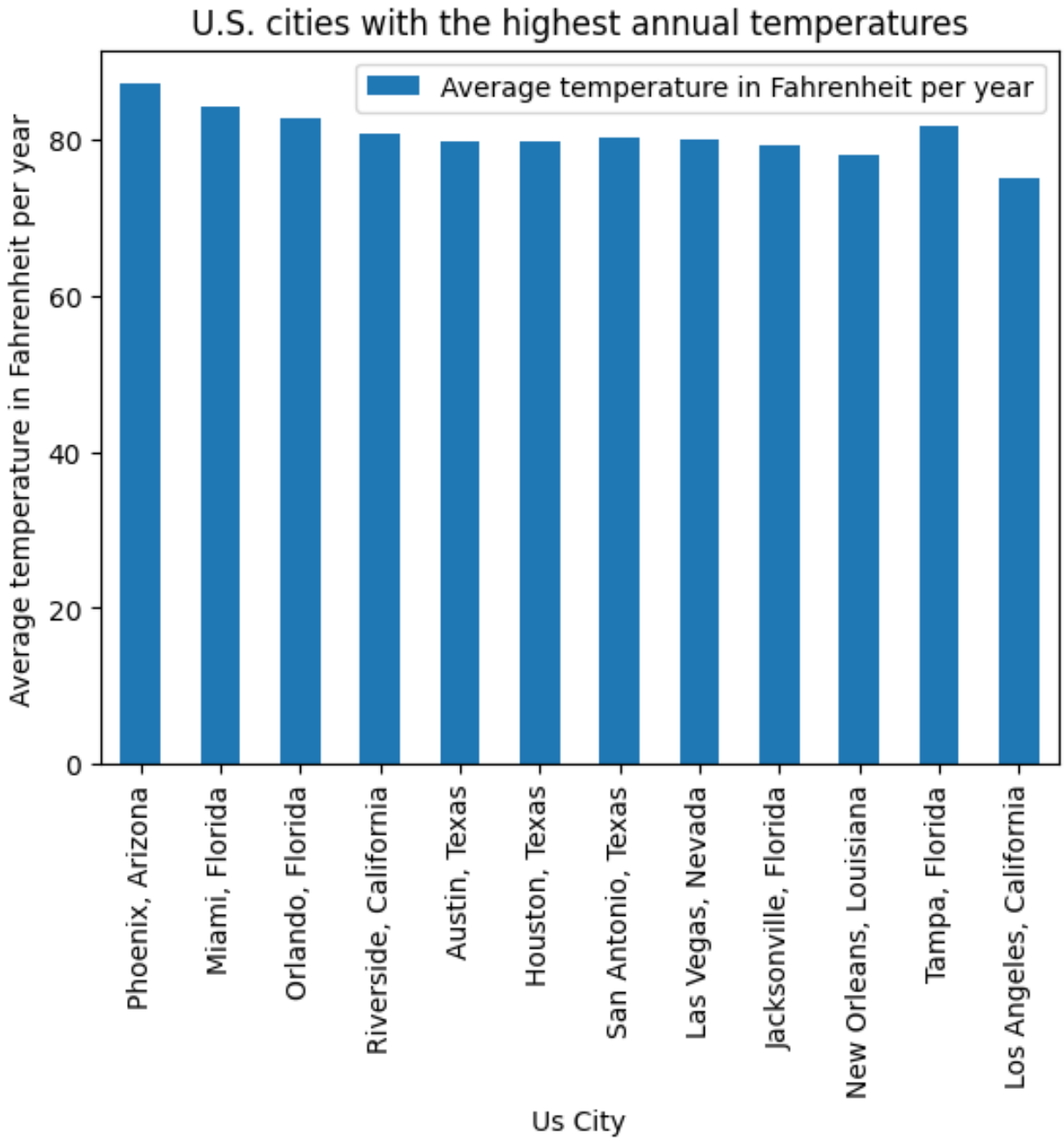


Figure 6.3: Summary by baseline: This statistic shows the top 25 cities in the United States with the highest average annual temperatures between January and August 2018. In August 2018, the average temperature in Phoenix, Arizona was 87.2 Fahrenheit per year.

Similarly, summaries from NLI+T5 and NLI+T5+280 were also incorrectly annotated due to hastiness. Another problems that evaluators must have faced was boredom. Evaluating 30 summaries in one experiment is incredibly boring. One of the evaluators said that they got bored after the second chart.

7. Conclusions

This chapter concludes the thesis. We talk about key takeaways from our study in Section 7.1. And in Section 7.2, we talk about possible future work that can be carried out to produce better summaries and reduce hallucinations in chart summarization.

7.1 Takeaways

Long-Distance Dependencies: There is no one right way to format the input data but the main point to keep in mind when deciding on the input format is to minimize long-term dependencies (see Section 5.1). Transformers are generally better at processing long sequences but even within those long sequences, dependency length can be a problem.

Providing More Context: Including title, x and y axis labels, and legends from the chart are important to generate faithful summaries. It gives the model more context at inference time. Otherwise, the generated summaries contains named entities from parametric knowledge instead of the input.

Training to Hallucinate: We showed in Section 5.2 that if training summaries contain information not grounded in the data, we get summaries that contain information not grounded in the data. When we use the training summaries with additional information, we are training our model to hallucinate. It is very important to keep summaries grounded in the input data. For pragmatic purposes, the summary can contain information that will spark interest in the reader, but we also need to understand that the model can overfit on the type of sentences that have nothing to do with the input data.

Chart Summarization Human Evaluation: Human evaluation of chart summarization task can be difficult for the annotators. We previously mentioned in Section 6.6 that annotators get confused when the value mentioned in the summary is not explicitly shown in the chart. Annotators also find this task incredibly boring because they have to read the same summary when evaluating for different properties. We felt that some annotators do not thoroughly read the summary again, and answered based on their first examination.

7.2 Possible Future Work

The main focus of future work should be to create a large scale dataset that contains analytical summaries grounded in the data such as in Table 7.1. Škrjanec et al. [2022] have created such a dataset but it is small, only containing 1063 summaries. Autochart [Zhu et al., 2021] contains analytical summaries, but those summaries are too formulaic, and often times the values in the summaries are direclty copied from the input data. In a summary, it is better to write the value '1,400,000' as '1.4 million' and not '1,400,000'.

To make the model produce analytical information, it is not enough to include such information in the chart summary. Analytical information should also be included in the linearized data input as the language models themselves are not well suited for numerical inference tasks [Andor et al., 2019, Neeraja et al., 2021, Chen et al., 2020b]. Computing values like maximum, average, and minimum is very straight forward. As shown in Table

7.1, this analytical information can be encoded in the linearized input and the language model will be able to learn more features from the data.

Data	|chart-data| Detroit Tigers all-time home run leaders 2019 x-y labels Players - Number of home runs, x-y values Al Kaline 399 , Norm Cash 373 , Miguel Cabrera 339 , Hank Greenberg 306 , Willie Horton 262 , Cecil Fielder 245 , Lou Whitaker 244 , Rudy York 239 , Lance Parrish 212 , Bill Freehan 200 |stats| maximum 399 , minimum 200 , average 281.9
Summary	This statistic shows the Detroit Tigers all-time home run leaders as of October 11 , 2019 . Al Kaline has hit the most home runs in Detroit Tigers franchise history with 399 home runs and Bill Freehan has hit the least home runs. On average, players have hit 281.9 home runs.

Table 7.1: Format for including maximum, minimum, and average in the input data.

Lastly, an important area to work on is to create a pre-trained NLI model that is specifically designed to check entailment in linearized data inputs. The NLI model we used in Chapter 6 was trained on the MultiNLI corpus [Williams et al., 2018], which only contains linguistic input. A model designed to check entailment of text in linearized data would help in the pre-processing stage as well as the evaluation. The importance of this model is not just specific to chart-to-text but all data-to-text tasks.

Bibliography

Jay Alammar. The illustrated transformer, 2018.

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. Giving BERT a calculator: Finding operations and arguments with reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5947–5952, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1609. URL `https://aclanthology.org/D19-1609`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Regina Barzilay and Mirella Lapata. Collective content selection for concept-to-text generation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 331–338, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics. URL `https://aclanthology.org/H05-1042`.

Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018.

Charles Chen, Ruiyi Zhang, Eunyee Koh, Sungchul Kim, Scott Cohen, and Ryan Rossi. Figure captioning with relation maps for reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020a.

Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. Logical natural language generation from open-domain tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7929–7942, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.708. URL `https://aclanthology.org/2020.acl-main.708`.

Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

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

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

Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W Cohen. Handling divergent reference texts when evaluating table-to-text generation. *arXiv preprint arXiv:1906.01081*, 2019.

Ondřej Dušek and Zdeněk Kasner. Evaluating semantic accuracy of data-to-text generation with natural language inference. *arXiv preprint arXiv:2011.10819*, 2020.

Nouha Dziri, Andrea Madotto, Osmar Zaiane, and Avishek Joey Bose. Neural path hunter: Reducing hallucination in dialogue systems via path grounding. *arXiv preprint arXiv:2104.08455*, 2021.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1213. URL `https://aclanthology.org/P19-1213`.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3518. URL `https://aclanthology.org/W17-3518`.

Li Gong, Josep Crego, and Jean Senellart. Enhanced transformer model for data-to-text generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 148–156, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5615. URL `https://aclanthology.org/D19-5615`.

Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. `http://www.deeplearningbook.org`.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland, December 2020. Association for Computational Linguistics. URL `https://aclanthology.org/2020.inlg-1.23`.

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

Yichong Huang, Xiachong Feng, Xiaocheng Feng, and Bing Qin. The factual inconsistency problem in abstractive text summarization: A survey. *arXiv preprint arXiv:2104.14839*, 2021.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *arXiv preprint arXiv:2202.03629*, 2022.

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

Hassan Kane, Muhammed Yusuf Kocyigit, Ali Abdalla, Pelkins Ajanoh, and Mohamed Coulibali. NUBIA: NeUral based interchangeability assessor for text generation. In *Proceedings of the 1st Workshop on Evaluating NLG Evaluation*, pages 28–37, Online

(Dublin, Ireland), December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.evalnlgeval-1.4.

Shankar Kanthara, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. Chart-to-text: A large-scale benchmark for chart summarization. *arXiv preprint arXiv:2203.06486*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*, 2017.

Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*, 2016.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. Hallucinations in neural machine translation, 2019. URL https://openreview.net/forum?id=SkxJ-309FQ.

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

Percy Liang, Michael Jordan, and Dan Klein. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL https://aclanthology.org/P09-1011.

Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology, 22 140, 55*, 1932.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. Table-to-text generation by structure-aware seq2seq learning. *arXiv preprint arXiv:1711.09724*, 2017.

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

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. *arXiv preprint arXiv:2109.05052*, 2021.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. Results of the wmt18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the third conference on machine translation: shared task papers*, pages 671–688, 2018.

Bill MacCartney. *Natural language inference*. Stanford University, 2009.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.173. URL https://aclanthology.org/2020.acl-main.173.

J. Neeraja, Vivek Gupta, and Vivek Srikumar. Incorporating external knowledge to enhance tabular reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2799–2809, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.224. URL https://aclanthology.org/2021.naacl-main.224.

Feng Nie, Jin-Ge Yao, Jinpeng Wang, Rong Pan, and Chin-Yew Lin. A simple recipe towards reducing hallucination in neural surface realisation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2673–2679, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1256. URL https://aclanthology.org/P19-1256.

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

Richard Yuanzhe Pang, Adam D Lelkes, Vinh Q Tran, and Cong Yu. Agreesum: Agreement-oriented multi-document summarization. *arXiv preprint arXiv:2106.02278*, 2021.

Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. Totto: A controlled table-to-text generation dataset. *arXiv preprint arXiv:2004.14373*, 2020.

Cécile L Paris, William R Swartout, and William C Mann. *Natural language generation in artificial intelligence and computational linguistics*, volume 119. Springer Science & Business Media, 2013.

Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, July 1900. doi: 10.1080/14786440009463897. URL https://doi.org/10.1080/14786440009463897.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319.

Xin Qian, Eunyee Koh, Fan Du, Sungchul Kim, Joel Chan, Ryan A Rossi, Sana Malik, and Tak Yeon Lee. Generating accurate caption units for figure captioning. In *Proceedings of the Web Conference 2021*, pages 2792–2804, 2021.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

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

Ajit Rajasekharan. T5 - a model that explores the limits of transfer learning, 2019.

Vikas Raunak, Arul Menezes, and Marcin Junczys-Dowmunt. The curious case of hallucinations in neural machine translation. *arXiv preprint arXiv:2104.06683*, 2021.

Clément Rebuffel, Marco Roberti, Laure Soulier, Geoffrey Scoutheeten, Rossella Cancelliere, and Patrick Gallinari. Controlling hallucinations at word level in data-to-text generation. *Data Mining and Knowledge Discovery*, 36(1):318–354, 2022.

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

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

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

Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *towards data science*, 6(12):310–316, 2017.

Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. Underreporting of errors in NLG output, and what to do about it. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 140–153, Aberdeen, Scotland, UK, August 2021. Association for Computational Linguistics. URL `https://aclanthology.org/2021.inlg-1.14`.

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

Hongmin Wang. Revisiting challenges in data-to-text generation with fact grounding. *arXiv preprint arXiv:2001.03830*, 2020.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. Towards faithful neural table-to-text generation with content-matching constraints. *arXiv preprint arXiv:2005.00969*, 2020.

L. Wilkinson. Statistical methods in psychology journals: Guidlines and explanations. *American Psychologist*, 54(8):594–604, 1999.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-1101`.

Xinnuo Xu, Ondřej Dušek, Verena Rieser, and Ioannis Konstas. AggGen: Ordering and aggregating while generating. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural*

Language Processing (Volume 1: Long Papers), pages 1419–1434, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.113. URL https://aclanthology.org/2021.acl-long.113.

Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*, 2019.

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

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

Iza Škrjanec, Muhammad Salman Edhi, and Vera Demberg. Barch: an english dataset of bar chart summaries. In *Proceedings of the Language Resources and Evaluation Conference*, pages 3552–3560, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.380.

A. Appendix

A.1 Hyperparameters

Parameter	Value
Maximum input length	1024
Maximum target length	512
Truncation	True
Padding	max_length
batch size	2
Optimizer	Weighted Adam [Kingma and Ba, 2014]
learning rate	3e-4
weight decay	0.01
Training Epochs for baseline	6
Training Epochs for NLI+T5	6
Training Epochs for NLI+T5+280	5
Training Epochs for model_t5_big	14
Beam size	4

A.2 Survey Description for Annotators

Dear Participants,
You will be evaluating summaries of charts. Choose the summary that has Value Correctness and Outside Information Presence. Rate the informativeness, coherence, and fluency of the summaries given the chart.
Value Correctness: Numbers/figures/values in the summary are from the chart. Here you determine which of the summaries are accurate
Outside Information: Information that is not from the chart at all. Here you determine which of the summaries have information not taken from the chart.
Informativeness: The summary conveys a lot of information about the chart. Here you rate the informativeness of the summary. 1 being the least informative and 5 being the most informative.
Coherence: The information included in the summary is orderly and logically consistent. Here you rate the coherence of the summary. 1 being the least coherent and 5 being the most coherent.
Fluency: Summary is grammatically correct and does not contain any repetitions. Here you rate the fluency of the summary. 1 being the least fluent and 5 being the most fluent.
When the participants open the survey, there is an example of how to fill the form and what to keep in mind when evaluating outside information. Kindly read and understand it thoroughly.

List of Figures

List of Tables