ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

School of Information Sciences and Technology
Department of Informatics
Athens, Greece

MSc in Data Science

Thesis

Training and Development of a Table-to-Text Transformer-Based Model for Contextual Summarization of Tabular Data

Despoina Angelonidi

Academic Supervisor: Dr. Panagiotis Louridas
Department of Management Science and Technology
Athens University of Economics and Business

Company Supervisor: Dr. Kostas Tsagkaris
Incelligent IKE

March 2024

Despoina Angelonidi

Training and Development of a Table-to-Text Transformer-Based Model for Contextual Summarization of Tabular Data

March 2024

Supervisor: Dr. Panagiotis Louridas

Athens University of Economics and Business

School of Information Sciences and Technology

Department of Informatics

Athens, Greece

Abstract

The term Table-to-Text refers to the process of converting information from structured tables into natural language text. This can be achieved by converting the tables to a text format and then using a sequence-to-sequence (seq2seq) model, which predicts the next token based on the context of the input and the previously generated tokens. On the spectrum of this project three transformer-based models are used, namely T5-small, T5-base [Raf+19] and Bart-base [Lew+19]. All models are trained on the ToTTo [Par+20] and QTSumm [Zha+23] datasets with the aim to generate targeted summaries that include the requested information.

Regarding ToTTo, the models are evaluated not only across the entire test set but also within the top 5 most popular domains. The findings suggest that the T5 variations exhibit strong performance in generating summaries for tables sourced from the Mixed Martial Arts Record category whereas the strength of the Bart-base model lies in generating summaries for tables within the domain of Demographics. Overall, the three models outperformed the benchmark. Concerning QTSumm, the models exhibited a similar level of performance to the benchmark across a wide range of metrics. The performance is relatively lower compared to ToTTo, which was anticipated given that QTSumm presents a more demanding task that requires more advanced reasoning abilities.

Περίληψη

Στη σημερινή εποχή ο όγκος των δεδομένων αυξάνεται συνεχώς όσο ποτέ άλλοτε. Ένα μεγάλο μέρος αυτών των δεδομένων είναι δομημένο σε μορφή πίνακα. Πολλές φορές η διάσταση των πινάκων είναι εκτενής, περιλαμβάνοντας πληροφορίες που δεν ενδιαφέρουν τον αναγνώστη. Δεδομένου ότι οι επιχειρήσεις αποσκοπούν στην εξοικονόμηση χρόνου και πόρων, υπάρχει η επιτακτική ανάγκη να αυτοματοποιηθούν όσες περισσότερες διαδικασίες είναι εφικτό. Σκοπός της παρούσας διπλωματικής εργασίας είναι η παραγωγή περιλήψεων γραμμένων σε φυσική γλώσσα όπου παρέχουν στον χρήστη την πληροφορία που αναζητά.

Για την παραγωγή των περιλήψεων εκπαιδεύτηκαν τρία μοντέλα σε δύο διαφορετικά datasets που υιοθετούν την αρχιτεκτονική των Transformers [Vas+17]. Συγκεκριμένα από την οικογένεια των T5 [Raf+19] επιλέχθηκαν το T5-small και το T5-base. Το τρίτο μοντέλο που χρησιμοποιήθηκε είναι το Bart-base [Lew+19].

Για την εκπαίδευση των μοντέλων, επιλέχθηκαν τα datasets ToTTo [Par+20] και QTSumm [Zha+23]. Στόχος του πρώτου είναι η παραγωγή μιας πρότασης η οποία περιλαμβάνει πληροφορία που περιέχεται σε υποδεδειγμένα κελιά. Αυτό έχει ως αποτέλεσμα να μειώνεται ο όγκος των περιττών πληροφοριών. Σκοπός του δεύτερου είναι η παραγωγή περιλήψεων μίας παραγράφου που απαντούν στο ερώτημα του χρήστη. Τα ερωτήματα μπορεί να περιλαμβάνουν απλές στοχευμένες περιλήψεις των πινάκων, συγκρίσεις μεταξύ τιμών, κα. Καθώς τα μοντέλα δέχονται τα δεδομένα σε μορφή κειμένου, οι πίνακες πριν δοθούν στα μοντέλα μετασχηματίστηκαν χρησιμοποιώντας τη μέθοδο των Chen et al. [Che+22].

Όσον αφορά το ToTTo, τα ευρήματα υποδηλώνουν ότι οι παραλλαγές του T5 είναι ικανές να παράξουν πολύ καλές περιλήψεις για πίνακες που προέρχονται από την κατηγορία "Mixed Martial Arts Record", ενώ το Bart-base υπερτερεί στη δημιουργία περιλήψεων για πίνακες που εμπίπτουν στην κατηγορία "Demographics". Συνολικά, τα τρία μοντέλα ξεπέρασαν το benchmark. Συνεχίζοντας με το QTSumm, τα αποτελέσματα φαίνεται να είναι παρόμοια με αυτά του benchmark. Συγκριτικά με το ToTTo, η απόδοση είναι χαμηλότερη, γεγονός που δεν προκαλεί εντύπωση καθώς το κείμενο που παράγεται είναι μεγαλύτερο σε έκταση και απαιτεί αυξημένο επίπεδο λογικής σκέψης.

Acknowledgements

First and foremost, I would like to thank Dr. Kostas Tsagkaris and the Incelligent team for their collaboration, guidance and provision of resources, which enriched the depth of this study. A special thanks goes to Nikos Spanos who was always eager to answer all of my questions, even beyond regular working hours. Lastly, I would like to express my appreciation to my family and friends for their constant encouragement and understanding during this academic journey. Their support has been the cornerstone of my perseverance.

Contents

1 Introduction

Artificial Intelligence (AI) is an interdisciplinary field, merging computer science, mathematics and cognitive science aiming to simulate human-like intelligence. It enables computers to solve tasks traditionally requiring human skills, such as problem-solving, learning, perception and language understanding. Deep learning, exemplified by the Transformer Architecture, has revolutionized AI by modeling complex relationships in extensive datasets in lesser time. With applications spanning a wide range of domains, AI profoundly influences the society, the economy and technology reshaping how we interact with information.

1.1 Motivation and Problem Statement

In an era where the value of data surpasses that of gold, the continuous surge in information, coupled with the world's vigorous pace, underscores the imperative to automate processes. Information manifests in diverse formats, with tables being the most prevalent data structure. Even though tables are a convenient way to organize and present data, when they get very extensive, locating the information of interest becomes a time-consuming task. Hence, it is crucial to find an effective way to summarize tabular data so that it can help the user obtain the required information, enabling quick decision-making.

1.2 Thesis Scope

The scope of this thesis is to train and develop a Table-to-Text Transformer-based model that is able to generate coherent and contextually relevant summaries for diverse types of tabular data. To achieve this, three Transformers are employed that are trained on the ToTTo dataset [Par+20] to provide a one-sentence, controlled description alongside with the QTSumm dataset [Zha+23], a collection of data that is designed to produce table summaries tailored to users' information needs. The models of interest are the small version of Google's T5 [Raf+19] with 60M parameters, the base version of the same model with 220 M parameters and Facebook's Bart-base [Lew+19] with 140M parameters.

As the aforementioned Transformers accept sequential data as input, in order to train them, it is crucial to transduce the tabular data into text format. To accomplish this, a template-based serialization method is employed as outlined by Chen et al. in 2022 [Che+22]. The evaluation of model performance is conducted using well-established metrics such as ROUGE [Lin04], SACREBLEU [Pos18] and METEOR [BL05]. To see how this approach compares to what has already been applied a model named NARRATABLE [Sán22] is employed as a benchmark which is a fine-tuned Bloom-560m [Sca+22] on the ToTTo dataset. To assess the performance of the models on the QTSumm dataset, a fine-tuned T5-large model is used as a benchmark, trained by the researchers that released the QTSumm paper [Zha+23]. The experimentation on two different datasets allows for a comprehensive evaluation of the proposed method's generalization across diverse data sources.

2 Backround and Related Work

2.1 Backround

2.1.1 Table-to-Text generation

Table-to-text generation refers to a Natural Language Processing (NLP) task where the goal is to automatically generate coherent and human-readable textual descriptions based on structured tabular data.

2.1.2 Table Summarization

Table summarization, a specific aspect of table-to-text generation, is the task of automatically generating concise and informative summaries from structured tabular data. The primary objective is to distill the essential information within a table into a coherent and readable form, providing a high-level overview without losing critical details. This task is particularly valuable in scenarios where large datasets or complex information need to be quickly understood and communicated.

2.1.3 Transformers

The transformer architecture concept was introduced in 2017 by Vaswani et al [Vas+17] and it has since become a foundational tool for various natural language processing (NLP) tasks and beyond. Just like recurrent neural networks (RNNs), transformers are designed to handle sequential input data such as natural language and execute tasks like text summarization and translation. However, in contrast to RNNs, transformers are capable of processing the entire input simultaneously. The attention mechanism gives the model the ability to focus on the most relevant parts of the input for each output. For example, when the data is natural language, the translator doesn't have to process the text one word at a time. This enables parallelized processing, surpassing the capacity of RNNs and leading to a reduction in training time. The Transformer architecture employs an encoder-decoder structure that does not depend on recurrence and convolutions for output generation. The input sequence is mapped by the encoder to a series of continuous representations. Subsequently, the decoder takes the encoder's output and the previous time step's decoder output to produce a new output sequence.

Below is provided a visual representation of the aforementioned architecture.

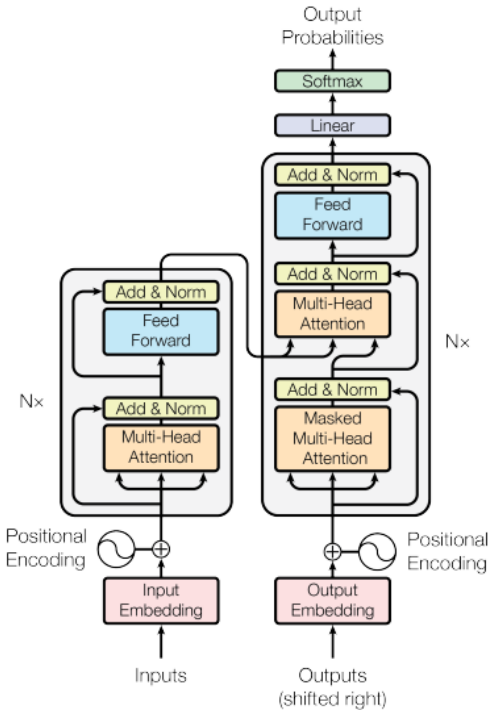


Fig. 2.1: An image that shows the Transformer - model architecture obtained by the original paper [Vas+17].

Since machines comprehend numerical data rather than text, this architecture initially converts the input into an n-dimensional embedding. Then, positional information is added to the embeddings using positional encoding, addressing the transformer's innate lack of understanding regarding the order or position of tokens within a sequence.

The encoder and decoder are comprised of many stacked modules, which mainly include feed-forward and multi-head attention layers. In the image above (Fig. 2.1) they are denoted as Nx.

As far as the Attention Mechanism is concerned, the equations of both "Scaled Dot-Product Attention", shown on the left side of Fig. 2.2, and "Multi-Head Attention" are provided below. In equation 2.1, Q represents a matrix containing the query, K consists of all Keys (the vector representations of all the words in the sequence) and V represents the Values.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \times V \tag{2.1}$$

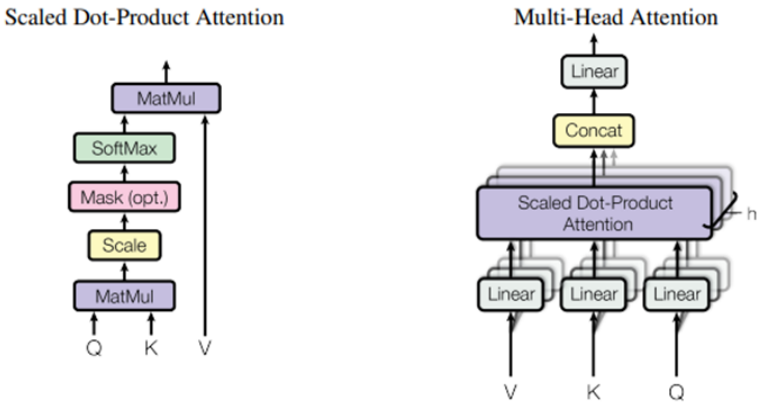


Fig. 2.2: An image obtained from the original paper [Vas+17] demonstrating Scaled Dot-Product and Multi-Head Attention.

On the right side of figure 2.2 there is a visual representation of Multi-Head Attention. Multi-Head Attention (Eq. 2.2) involves projecting input queries, keys, and values into multiple parallel heads, each with learned linear transformations. The attention function is then performed independently on each head, producing distinct output values. These outputs are concatenated and linearly projected to generate the final result. This approach enables the model to simultaneously focus on varied information from distinct representation subspaces, enhancing its ability to capture complex patterns and relationships in the data.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) \times \mathbf{W}_O \tag{2.2}$$

$$\text{where head}_i = \text{Attention}(Q\mathbf{W}_i^Q, K\mathbf{W}_i^K, V\mathbf{W}_i^V)$$

Here, $\mathbf{W}_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, $W^O \in \mathbb{R}^{hd_{\text{v}} \times d_{\text{model}}}$ are the parameter matrices of the projections.

But where is attention encountered?

- In the Encoder where the input sequence pays attention to itself (Self-attention).
- In the Decoder where the target sequence pays attention to itself (Self-attention).
- In the Decoder where the target sequence pays attention to the input sequence (Encoder-Decoder-attention).

2.2 Related Work

2.2.1 Early Work

Inspired by the promising results of Sequence-to-Sequence (seq2seq) techniques in machine translation and text summarization, Lebret et al. in 2016 [LGA16] and Wiseman et al. in 2017 [WSR17] suggested to convert the input table into a sequence of records. Next, Liu et al in 2017 [Liu+17] and Gong et al. in 2019 [Gon+19], aiming to enhance the effectiveness of table-to-text methods using seq2seq, encoded not only the context of the table but also its structure.

2.2.2 Table-to-Text Generation with Transformer Models

In addition, recent Transformers were also employed to solve the table-to-text task. As these models are designed to handle textual data as input, the tables were converted into textual sequences before processing.

In 2020, Mihir Kale and Abhinav Rastogi [KR20] studied the pre-training and fine-tuning strategy for data-to-text tasks. Their experiments demonstrated that utilizing text-to-text pre-training, exemplified by T5 [Raf+19], empowers straightforward end-to-end transformer models to surpass the performance of pipelined neural architectures specifically designed for data-to-text generation. The results also indicated that T5 [Raf+19] pre-training led to better generalization, as evidenced by large improvements on out-of-domain test sets.

Introduced in 2022 by Ewa Andrejczuk et al. [And+22], TABT5 (Table-and-Text-to-Text Transfer Transformer) is an encoder-decoder model that generates natural language text based on tables and textual inputs using the pre-trained T5 transformer [Raf+19] as a baseline architecture. Each table is linearized into a sequence of words, and these words are split into tokens. Next, the input sequence is created by concatenating the question and table tokens. In order to encode the table structure, row and column embeddings are included in the model, added on top of the token embeddings as inputs, and optimized during training.The target sequence generated by the model can be a free-form answer, serving various purposes such as answering questions, providing table summaries without specific questions, or generating formulas for formula prediction tasks. Two pre-training strategies were used. The first strategy, Denoising, involves training the model to predict a target sequence containing the missing or corrupted tokens in the input table. By replacing 15% of cells and columns with mask tokens the model becomes more capable when it comes to capturing relationships between neighboring cells and related text. The second strategy implemented after Denoising, ToTTification, is inspired by ToTTo [Par+20] and involves

retrieving statements related to Wikipedia tables, using them as target text for pre-training. Matching entities in these statements are added to the input to guide generation. The model was pre-trained using the dataset proposed by Herzig et al. [Her+20] in 2020. For evaluation 4 datasets were used, namely WIKISQL introduced by Zhong et al. in 2017, [ZXS17], ENRON by Chen et al. released in 2021 [Che+21a], TOTTO by Parikh et al. introduced in 2020 [Par+20] and FINQA by Chen et al. released in 2021 [Che+21b].

BLOOM, introduced by Teven Le Scao et al. in 2022 [Sca+22], is a decoder-only Transformer language model that was trained on the ROOTS corpus [Lau+22], a dataset comprising hundreds of sources in 46 natural and 13 programming languages. There are 6 variations available so far with the lightest version, BLOOM-560M, having 559 hyperparameters and 16 attention heads. Upon BLOOM's release, in the same year, the team of Narrativa, led by Manuel Romero, released NARRATABLE [Sán22]. This is an open-source model which uses BLOOM's smallest version as a base and is trained on the ToTTo dataset [Par+20]. This model, which is also the benchmark of this project regarding training performed using the ToTTo dataset, generates text from tables that include only the information of interest saving the user time and energy. The input tables are passed to the model after being serialized adopting the following format (Tab. 2.1):

<s><page_title> John Higgins </page_title> <section$_{t}$itle > $Minor - rankingfinals : 6(3titles, 3runners - up) <$ $/section_{t}itle >< table >< row >< c > Outcome < /c >< c > No. < row_header > Outcome <$ $/row_header >< /c >< c > Year < row_header > Outcome < /row_header >< row_header > No. <$ $/row_header >< /c >< c > Championship < row_header > Outcome < /row_header >< row_header >$ $No. < /row_header >< row_header > Year < /row_header >< /c >< c > Opponentinthefinal <$ $row_header > Outcome < /row_header >< row_header > No. < /row_header >< row_header > Year <$ $/row_header >< row_header > Championship < /row_header >< /c >< c > Score < row_header >$ $Outcome < /row_header >< row_header > No. < /row_header >< row_header > Year < /row_header ><$ $row_header > Championship < /row_header >< row_header > Opponentinthefinal < /row_header ><$ $/c >< /row >< row >< c > Winner < col_header > Outcome < /col_header >< /c >< c > 1. <$ $col_header > No. < /col_header >< /c >< c > 2010 < col_header > Year < /col_header >< /c >< c >$ $RuhrChampionship < col_header > Championship < /col_header >< /c >< c > EnglandShaunMurphy <$ $col_header > Opponentinthefinal < /col_header >< /c >< c > 4ˇ2 < col_header > Score < /col_header ><$ $/c >< /row >< row >< c > Runner - up < col_header > Outcome < /col_header >< /c >< c > 1. <$ $col_header > No. < /col_header >< /c >< c > 2010 < col_header > Year < /col_header >< /c ><$ $c > PragueClassic < col_header > Championship < /col_header >< /c >< c > EnglandMichaelHolt <$ $col_header > Opponentinthefinal < /col_header >< /c >< c > 3ˇ4 < col_header > Score < /col_header ><$ $/c >< /row >< row >< c > Runner - up < col_header > Outcome < /col_header >< /c >< c >$ $2. < col_header > No. < /col_header >< /c >< c > 2011 < col_header > Year < /col_header ><$ $/c >< c > PlayersTourChampionshipˇEvent5 < col_header > Championship < /col_header >< /c ><$ $c > EnglandAndrewHigginson < col_header > Opponentinthefinal < /col_header >< /c >< c > 1ˇ4 <$ $col_header > Score < /col_header >< /c >< /row >< row >< c > Winner < col_header > Outcome <$ $/col_header >< /c >< c > 2. < col_header > No. < /col_header >< /c >< c > 2012 < col_header > Year <$ $/col_header >< /c >< c > KaySuzanneMemorialTrophy < col_header > Championship < /col_header ><$ $/c >< c > EnglandJuddTrump < col_header > Opponentinthefinal < /col_header >< /c >< c > 4ˇ2 <$ $col_header > Score < /col_header >< /c >< /row >< row >< c > Runner - up < col_header > Outcome <$ $/col_header >< /c >< c > 3. < col_header > No. < /col_header >< /c >< c > 2012 < col_header >$ $Year < /col_header >< /c >< c > BulgarianOpen < col_header > Championship < /col_header ><$ $/c >< c > EnglandJuddTrump < col_header > Opponentinthefinal < /col_header >< /c >< c > 0ˇ4 <$ $col_header > Score < /col_header >< /c >< /row >< row >< highlighted_cell > Winner < col_header >$ $Outcome < /col_header >< /highlighted_cell >< c > 3. < col_header > No. < /col_header >< /c ><$ $highlighted_cell > 2013 < col_header > Year < /col_header >< /highlighted_cell >< highlighted_cell >$ $BulgarianOpen < col_header > Championship < /col_header >< /highlighted_cell >< highlighted_cell >$ $AustraliaNeilRobertson < col_header > Opponentinthefinal < /col_header >< /highlighted_cell ><$ $highlighted_cell > 4ˇ1 < col_header > Score < /col_header >< /highlighted_cell >< /row >< /table >$

Tab. 2.1: Example input used in NARRATABLE's inference API section included in its corresponding Huggingface page.

Here special tokens are introduced to pass the table structure information to the model. This method was previously implemented by Clayton Leroy Chapman et al in 2021 [Cha+21] and many others. Prior to BLOOM [Sca+22], NARRATIVA trained the T5-base on the same

dataset and task. However, due to the limitations of T5 variations [Raf+19], which can only encode 512 tokens, the model couldn't take the whole serialized table as an input. As a result, it couldn't effectively answer the user's questions according to NARRATIVA [Sán22]. For this reason they employed BLOOM [Sca+22] which has a wider context window. To test the models, they used the original ToTTo evaluation set (7700 examples) as a test set.

In 2023, Yilun Zhao et al. [Zha+23], introduced the QTSumm dataset and evaluated 11 different models. To be specific they fine-tuned the large versions of Bart [Lew+19], T5 and Flan-T5 [Chu+22]. They also evaluated on QTSumm three state-of-the-art Table-to-Text Generation Models. These are TAPEX [Liu+21], REASTAP [Zha+22] and OmniTab [Jia+22]. To provide the data to the models, the tables got serialized by flattening. The flattening process involved representing the table data as:

$$T = [HEADER] : h, [ROW]_1 : r1, ..., [ROW]_n : rn \tag{2.3}$$

where 'h' represents the table header, and 'r_i' is the i-th table row. Special tokens ([HEADER] and [ROW]) were used to denote table headers and rows. In text generation models, these tokens indicated specific regions, while in Language Models (LLMs), they were set as empty strings. Headers or cells in different columns were separated using a vertical bar (|). This flattened format allowed direct input into text generation models. For table-to-text generation models, the authors adhered to the original data processing methods for inputting both the user query and table data.

Continuing with the models, TAPEX [Liu+21] is an execution-focused table pre-training method which generates its corpus by automatically synthesizing samples of SQL queries along with their execution results. In the encoding process, the natural language sentence is directly tokenized, while the structured table undergoes a flattening procedure using special tokens. This flattened representation, includes indicators for headers, rows, and column separations. The natural language sentence and flattened table are then concatenated and fed into the model encoder for downstream processing. During decoding, the model generates answers or decisions autoregressively. For Table Question Answering, it produces answers word by word for given natural language questions. For Table-based Fact Verification, a binary classifier, based on the decoder's last token, determines if the natural language sentence aligns with facts in the associated table. While using BART [Lew+19] as backbone, TAPEX outperforms prior methods in Table Question Answering and Table-based Fact Verification tasks, achieving state-of-the-art results with even small pre-training data.

The table pre-training approach of REASTAP [Zha+22] demonstrates that advanced table reasoning abilities can be incorporated into models during pre-training without the need for any complex, table-specific architecture design. In this approach diverse pre-defined table reasoning skills are injected into the models by training them to generate accurate

answers to synthetic questions. In contrast to prior approaches that involve designing table-specific architectures, REASTAP is straightforward to implement and theoretically applicable to any sequence-to-sequence language model. To conduct their experiments, researchers implemented REASTAP on Bart-large [Lew+19]. The results showed that REASTAP obtains slightly lower results on BLEU scores compared to the BART backbone [Lew+19]. This is reasonable since they continued pre-training REASTAP on their pre-training corpus that appears to be irrelevant to the text generation task. Nonetheless, REASTAP notably enhances logical-fidelity scores, boosting Parsing-based Evaluation and NLI-based Evaluation, two metrics introduced by Chen et al. [Che+20] in 2020 .

OmniTab [Jia+22] is an end-to-end model requiring minimal annotated natural language questions. Leveraging natural and synthetic pre-training data, OmniTab aligns natural language with tables using retrieval-based methods and generates synthetic natural language questions from sampled SQL queries, showcasing its effectiveness with state-of-the-art performance in few-shot settings when combining natural and synthetic pre-training.

Continuing with QTSumm Experiments, each model was fine-tuned for 15 epochs with a batch size of 128 and the best fine-tuning checkpoints were selected by taking the validation loss into account. Finally they performed zero-shot, 1-shot and 2-shot training on Llama2 [Tou+23], Vicuna [Zhe+23], Mistral [Jia+23], Lemur [Xu+23] and GPT [Ach+23]. For the open-sourced LLMs the temperature was set to 1, the top P to 1 and the maximum output length to 256. The results showed that the Table-To-Text generation models outperformed their corresponding backbones,highlighting the importance that table structure holds regarding this task. Among text generation models Flan-T5 outperformed T5 and LLMs with improved reasoning capabilities such as Llama-2-70B and GPT-4 also achieve better performance. Finally, GPT models exhibit better performance than state-of-the-art fine-tuned models in human evaluation.

Since transformer models are capable of understanding context and generating text focusing on important information, in this thesis a new way of introducing table structure to the models and highlighting important information is proposed by converting the input table into a text template format including parts such as "Focus on:" and "Table Structure: Number of Rows: x, Number of Columns: y", where x and y denote number of table rows and columns accordingly. Finally, by employing the ToTTo dataset as well as the QTSumm dataset and including the instruction "Summarize:" to the beginning of each template for ToTTo and the corresponding query for QTSumm, targeted text summaries are generated including the requested information instead of summarizing the entire table. While this approach could be considered simplistic, the results suggest that the models outperformed NARRATABLE and performed close to some of the transformers and LLMs fine-tuned and tested in the QTSumm paper [Zha+23].

3 Dataset Description

3.1 ToTTo

The first dataset employed in this project, ToTTo [Par+20], is an open-domain collection consisting of tables gathered from Wikipedia across diverse domains, including science, sports, geography, demographics, films, and more. As the pie chart provided in the dataset paper denotes (Fig. 3.1), the majority of the tables belong to the "Sports" category followed by "Countries" and "Performing Arts".

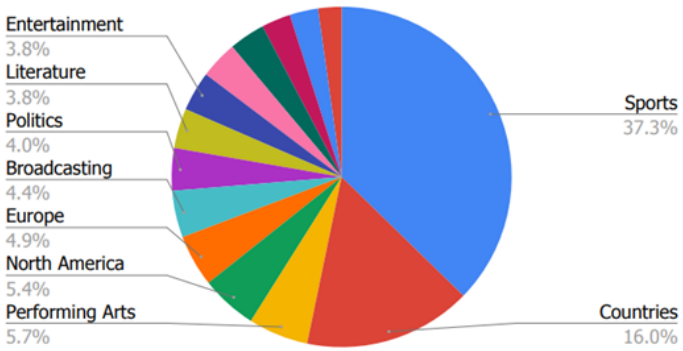


Fig. 3.1: This image, which originates from the paper of ToTTo [Par+20], illustrates the distribution of tables across diverse domains.

Each table of the dataset is paired with a concise one-sentence description and pertinent metadata. Notably, ToTTo introduces a distinctive feature known as "highlighted cells," indicating cells that contain crucial information found in the corresponding annotation sentence. By leveraging this information, the model of choice and, by extension, the user is able to generate guided text summaries. The dataset is composed of:

- the table itself
- the table webpage URL.
- a unique ID for each example
- the name of the page from which the table originates (e.g. “Alma Jodorowsky”)
- the title of the section that the table belongs to (e.g. “Demographics”, “Film”, etc.)
- a brief text regarding the table section (e.g. “Parties Democratic Republican”)
- the highlighted cells (a list of lists that include the coordinates of the cells)

- sentence annotations that encompass the original sentence and the series of revised sentences conducted sequentially to generate the final sentence. In this project, only the final sentence will be used as an annotation.
- a column that states whether the set of interest features examples that are out of domain from the training set.

When it comes to the datatype of each field, the majority is in string format. It is necessary to explain the structure of the Table field to facilitate the readers' understanding at the stage of preprocessing. The tables come in a table (List[List[Dict]]) format where the outer lists represent rows and the inner lists represent columns.

The original training set of ToTTo includes 120,761 tables. For this project, the training set constitutes the initial 90% of the original, with the remaining 10% serving as a validation set. The original validation set is subsequently employed as the test set. The reasoning behind this action is that the ToTTo dataset does not openly provide the annotations for the original test set.

New Split		
Training Set	Validation Set	Test Set
108,685	12,076	7,700

Tab. 3.1: The New Split of ToTTo dataset [Par+20] used in this project.

It is important to mention that the original validation set (here used as a test set) contains 3,916 tables that are out-of-domain from the original training set and by extension out-of-domain from the training and validation set employed in this project. This makes the task more challenging as the models have to provide summaries for tables that come from domains that they haven't encountered during training.

3.2 QTSumm

QTSumm is a Table-to-Text generation dataset, proposed by Yilun Zhao et al. in 2023 [Zha+23], that incorporates 7,111 query-summary pairs over 2,934 Wikipedia tables covering diverse topics. Fig. 3.2, obtained from the paper of the dataset, provides an insight regarding the domains that are covered. The majority of tables belongs to the "Sports" domain followed by "Statistics" and "Celebrity".

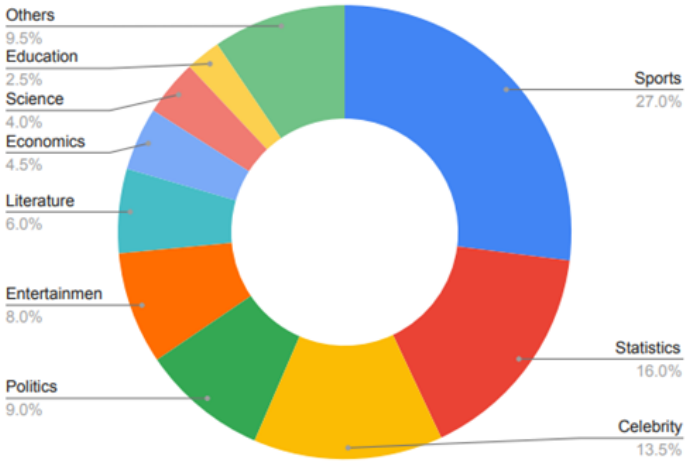


Fig. 3.2: A plot obtained from the QTSumm paper [Zha+23] demostrating the percentage of tables per domain.

This dataset is designed to facilitate the training of models for summarization tasks with a query-oriented approach. Here instead of highlighted cells, highlighted rows are provided. Those are the rows of each table that include the information contained in the human-generated summaries. The dataset is comprised by:

- the table itself
- the human-generated summary
- the query
- an example ID
- and the highlighted rows

Tab. 3.2 showcases the way that the QTSumm dataset is being split. The training set consists of 4,981 examples, the validation set is composed of 1,052 examples and the test set includes 1,078 examples.

Split		
Training Set	**Validation Set**	**Test Set**
4,981	1,052	1,078

Tab. 3.2: Split of QTSumm dataset.[Zha+23]

Tab. 3.3 includes some query examples along with their corresponding reference summaries. It is worth mentioning that the queries vary in terms of requests and seem to command higher reasoning capabilities.

Query	Annotation
Do Australian rules football matches receive higher attendance than cricket matches when analyzing the top 10 attendance records at the Bellerive Oval?	On the basis of top 10 records of attendance at Bellerive Oval, it seems cricket matches have more people attending than Australian rules football game. In top 10 match, 7 of them cricket matches, only 3 of them Australian rules football game. Plus, highest record of attendance for cricket match have 18,149 person, but highest for Australian rules football is 17,844 person.
What is the relationship between the percentage of seats won and the party's position as an opposition or the majority government in the given years?	The percent of seats win by the National Democratic Party (British Virgin Islands) direct influence party's position as opposition or majority government in the give years. When party win higher percent of seats (52.4% in 2003, 52.5% in 2011, and 60.2% in 2015), they have majority government. But, when their percent of seats lower (37.3% in 1999, 39.6% in 2007, and not know amount in 2019), they be opposition.
Compare the electricity production from wind power in Germany and Japan in 2011.	In 2011, Germany produced 45.3 terawatt-hours of electricity from wind power, and Japan produced 4.35 terawatt-hours of electricity from wind power.
Summarize the basic information of the wives of Amasa Lyman that had no children.	Amasa Lyman married one women who had no children. Diontha Walker, age 27 when she married Lyman in 1843, was the women with no children.
What was the range of attendances seen at events at The Pearl at The Palms venue in 2009?	the range of attendances seen at events at The Pearl at The Palms venue in 2009 was from 1,741 to 13,027.

Tab. 3.3: QTSumm: Query - Annotation pair examples

Data Preprocessing

4

In both datasets, converting tables to text format is essential for providing data to the models. Subsequently, the following section outlines the preprocessing steps employed to generate the text templates.

4.1 ToTTo

During the preprocessing stage, it was observed that some cells were absent. The following example shows that the first list is composed of 4 dictionaries (cells), while the second list consists of 8 dictionaries. To facilitate the process of data manipulation, each list needs to have the same number of dictionaries so that the arrays can be converted to dataframes. In this example, each list must contain a total of 9 cells. Since the first subarray includes a dictionary with a row span of 2, it is necessary to include this dictionary in the second subarray. This indicates that the value of the first cell also appears in the cell directly below it.

Concerning column span, a value greater than one implies that the cell's (dictionary's) value of interest is also present in other consecutive cells in the same row. For instance, when the column span is equal to three, the value of this dictionary must appear in three consecutive cells of the same row, as this value spans 3 columns. So, in the first subarray the dictionary that contains the value "Designation" and the dictionary that contains the value "Discovery" must be added two more times in this subarray. Following the same chain of thought, the dictionary that includes the value "Properties" must be added once more.

A snippet of a table with column span and row span greater than one.

```
array([{'column_span':3, 'is_header':True, 'row_span':1, 'value':'Designation'},
        {'column_span':3, 'is_header':True, 'row_span':1, 'value':'Discovery'},
        {'column_span':2, 'is_header':True, 'row_span':1, 'value':'Properties'},
        {'column_span':1, 'is_header':True, 'row_span':2, 'value':'Ref'}],
        dtype=object)
,
array([{'column_span':1, 'is_header':True, 'row_span': 1, 'value':'Permanent'},
       {'column_span':1, 'is_header':True, 'row_span':1, 'value':'Provisional'},
       {'column_span':1, 'is_header':True, 'row_span':1, 'value':'Citation'},
       {'column_span':1, 'is_header':True, 'row_span':1, 'value':'Date'},
       {'column_span':1, 'is_header':True, 'row_span':1, 'value':'Site'},
       {'column_span':1, 'is_header':True, 'row_span':1, 'value':'Discoverer'},
       {'column_span':1, 'is_header':True, 'row_span':1, 'value':'Category'},
       {'column_span':1, 'is_header':True, 'row_span':1, 'value':'    '}],
       dtype=object)
```

As the models employed operate on textual inputs, it is imperative to transform tables into text format. This conversion process adheres to the "Header is Value" pattern. To illustrate, consider the table portrayed in Fig. 4.1, which presents the highest attendances during the 2014 Indian Super League season.

Rank	Home team	Score	Away team	Attendance	Date	Stadium
1	Atlético de Kolkata	3–0	Mumbai City	65,000	12 October 2014	Salt Lake Stadium
2	Kerala Blasters	0–1	Chennaiyin FC	61,323	30 November 2014	Jawaharlal Nehru Stadium
3	Kerala Blasters	3–0	Chennaiyin FC	60,900	13 December 2014	Jawaharlal Nehru Stadium
4	Kerala Blasters	2–1	Atlético de Kolkata	57,296	21 November 2014	Jawaharlal Nehru Stadium
5	Atlético de Kolkata	1–1	Delhi Dynamos	55,793	19 October 2014	Salt Lake Stadium
6	Atlético de Kolkata	0–0	Goa	53,173	14 December 2014	Salt Lake Stadium
7	Kerala Blasters	1–0	Goa	49,517	6 November 2014	Jawaharlal Nehru Stadium
8	Atlético de Kolkata	0–0	Chennaiyin FC	46,288	14 November 2014	Salt Lake Stadium
9	Kerala Blasters	1–0	Pune City	44,532	9 December 2014	Jawaharlal Nehru Stadium
10	Kerala Blasters	0–0	NorthEast United	43,299	4 December 2014	Jawaharlal Nehru Stadium

Fig. 4.1: A Wikipedia table showing the highest attendances in the 2014 Indian Super League season.

By adopting the logic described previously, the table of Fig. 4.1 becomes:

Rank is 1, Home team is Atlético de Kolkata, Score is 30, Away team is Mumbai City, Attendance is 65,000, Date is 12 October 2014, Stadium is Salt Lake Stadium. Rank is 2, Home team is Kerala Blasters, Score is 01, Away team is Chennaiyin, Attendance is 61,323, Date is 30 November 2014, Stadium is Jawaharlal Nehru Stadium. Rank is 3, Home team is Kerala Blasters, Score is 30, Away team is Chennaiyin, Attendance is 60,900, Date is 13 December 2014, Stadium is Jawaharlal Nehru Stadium. Rank is 4, Home team is Kerala Blasters, Score is 21, Away team is Atlético de Kolkata, Attendance is 57,296, Date is 21 November 2014, Stadium is Jawaharlal Nehru Stadium. Rank is 5, Home team is Atlético de Kolkata, Score is 11, Away team is Delhi Dynamos, Attendance is 55,793, Date is 19 October 2014, Stadium is Salt Lake Stadium. Rank is 6, Home team is Atlético de Kolkata, Score is 00, Away team is Goa, Attendance is 53,173, Date is 14 December 2014, Stadium is Salt Lake Stadium. Rank is 7, Home team is Kerala Blasters, Score is 10, Away team is Goa, Attendance is 49,517, Date is 6 November 2014, Stadium is Jawaharlal Nehru Stadium. Rank is 8, Home team is Atlético de Kolkata, Score is 00, Away team is Chennaiyin, Attendance is 46,288, Date is 14 November 2014, Stadium is Salt Lake Stadium. Rank is 9, Home team is Kerala Blasters, Score is 10, Away team is Pune City, Attendance is 44,532, Date is 9 December 2014, Stadium is Jawaharlal Nehru Stadium. Rank is 10, Home team is Kerala Blasters, Score is 00, Away team is NorthEast United, Attendance is 43,299, Date is 4 December 2014, Stadium is Jawaharlal Nehru Stadium.

Fig. 4.2: The table of Fig. 4.1 converted to text.

Since there are also available metadata about the table, they will be included in the text in the following order: Table page title, Table section title, and Table section text. Next, the information of the highlighted cells is incorporated by adding a “Focus on:” part to convey

the important information to the model. Lastly, to provide information about the table structure, the number of rows and columns is added.

So, the final input template for table in Fig. 4.1 has the form:

Table page title: 2014 Indian Super League season, Table section title: Highest attendances, Table section text: Source:, Table converted to text: Rank is 1, Home team is Atlético de Kolkata, Score is 30, Away team is Mumbai City, Attendance is 65,000, Date is 12 October 2014, Stadium is Salt Lake Stadium. Rank is 2, Home team is Kerala Blasters, Score is 01, Away team is Chennaiyin, Attendance is 61,323, Date is 30 November 2014, Stadium is Jawaharlal Nehru Stadium. Rank is 3, Home team is Kerala Blasters, Score is 30, Away team is Chennaiyin, Attendance is 60,900, Date is 13 December 2014, Stadium is Jawaharlal Nehru Stadium. Rank is 4, Home team is Kerala Blasters, Score is 21, Away team is Atlético de Kolkata, Attendance is 57,296, Date is 21 November 2014, Stadium is Jawaharlal Nehru Stadium. Rank is 5, Home team is Atlético de Kolkata, Score is 11, Away team is Delhi Dynamos, Attendance is 55,793, Date is 19 October 2014, Stadium is Salt Lake Stadium. Rank is 6, Home team is Atlético de Kolkata, Score is 00, Away team is Goa, Attendance is 53,173, Date is 14 December 2014, Stadium is Salt Lake Stadium. Rank is 7, Home team is Kerala Blasters, Score is 10, Away team is Goa, Attendance is 49,517, Date is 6 November 2014, Stadium is Jawaharlal Nehru Stadium. Rank is 8, Home team is Atlético de Kolkata, Score is 00, Away team is Chennaiyin, Attendance is 46,288, Date is 14 November 2014, Stadium is Salt Lake Stadium. Rank is 9, Home team is Kerala Blasters, Score is 10, Away team is Pune City, Attendance is 44,532, Date is 9 December 2014, Stadium is Jawaharlal Nehru Stadium. Rank is 10, Home team is Kerala Blasters, Score is 00, Away team is NorthEast United, Attendance is 43,299, Date is 4 December 2014, Stadium is Jawaharlal Nehru Stadium, Focus on: Atlético de Kolkata 30 Mumbai City 12 October 2014 Salt Lake Stadium, Table Structure: 11 rows, 7 columns.

Fig. 4.3: Table template of Fig. 4.1 after incorporating the available metadata along with the information of the highlighted cells and table structure.

The table-to-text conversion is a challenging task as the dataset contains tables with a variety of formats that need different handling. For example, the table portrayed in Fig. 4.4 includes merged cells. In this case the header needs to be a combination of the first two rows.

Name	Command/ Response	Description	Info	C-Field Format							
				7	6	5	4	3	2	1	0
Set normal response mode **SNRM**	C	Set mode	Use 3 bit sequence number	1	0	0	P	0	0	1	1
SNRM extended **SNRME**	C	Set mode; extended	Use 7 bit sequence number	1	1	0	P	1	1	1	1
Set asynchronous response mode **SARM**	C	Set mode	Use 3 bit sequence number	0	0	0	P	1	1	1	1
SARM extended **SARME**	C	Set mode; extended	Use 7 bit sequence number	0	1	0	P	1	1	1	1
Set asynchronous balanced mode **SABM**	C	Set mode	Use 3 bit sequence number	0	0	1	P	1	1	1	1
SABM extended **SABME**	C	Set mode; extended	Use 7 bit sequence number	0	1	1	P	1	1	1	1
Set Mode **SM**	C	Set mode, generic	New in ISO 13239	1	1	0	P	0	0	1	1
Set initialization mode **SIM**	C	Initialize link control function in the addressed station		0	0	0	P	0	1	1	1

Fig. 4.4: A snippet of a Wikipedia table taken from the page "High-Level Data Link Control"

Following the completion of all requisite adjustments, the table depicted in Fig. 4.4 is now presented as follows:

'Table page title: High-Level Data Link Control, Table section title: Unnumbered frames, Table section text:, Table converted to text: Name is Set normal response mode SNRM, Com mand Response is C, Description is Set mode, Info is Use 3 bit sequence number, **C-Field Format 7** is 1, **C-Field Format 6** is 0, **C-Field Format 5** is 0, C-Field Format 4 is P, C-Field Format 3 is 0, C-Field Format 2 is 0, C-Field Format 1 is 1, C-Field Format 0 is 1. Name is SNRM extended SNRME, Command Response is C, Description is Set mode extended, Info i s Use 7 bit sequence number, C-Field Format 7 is 1, C-Field Format 6 is 1, C-Field Format 5 is 0, C-Field Format 4 is P, C-Field Format 3 is 1, C-Field Format 2 is 1, C-Field Form at 1 is 1, C-Field Format is 1. Name is Set asynchronous response mode SARM, Command Re sponse is C, Description is Set mode, Info is Use 3 bit sequence number, C-Field Format 7 is 0, C-Field Format 6 is 0, C-Field Format 5 is 0, C-Field Format 4 is P, C-Field Format 3 is 1, C-Field For...'

Fig. 4.5: A snippet of the template made for the table of Fig. 4.4 where cells need to be merged and used together as column header.

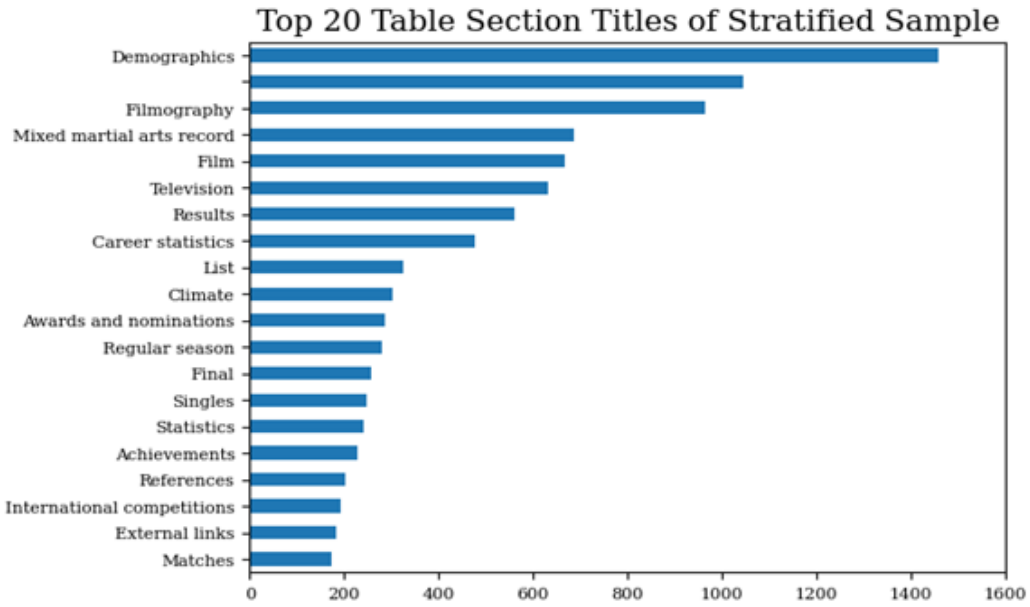
Notice that the last header is not just "C-Field Format" but "C-Field Format 7", "C-Field Format 6", "C-Field Format 5", etc.

Since the dataset comprises of thousands of tables it would be impossible to identify and take into consideration every single table format. Thus, an extensive effort has been made to create tailored templates for as many cases as possible.

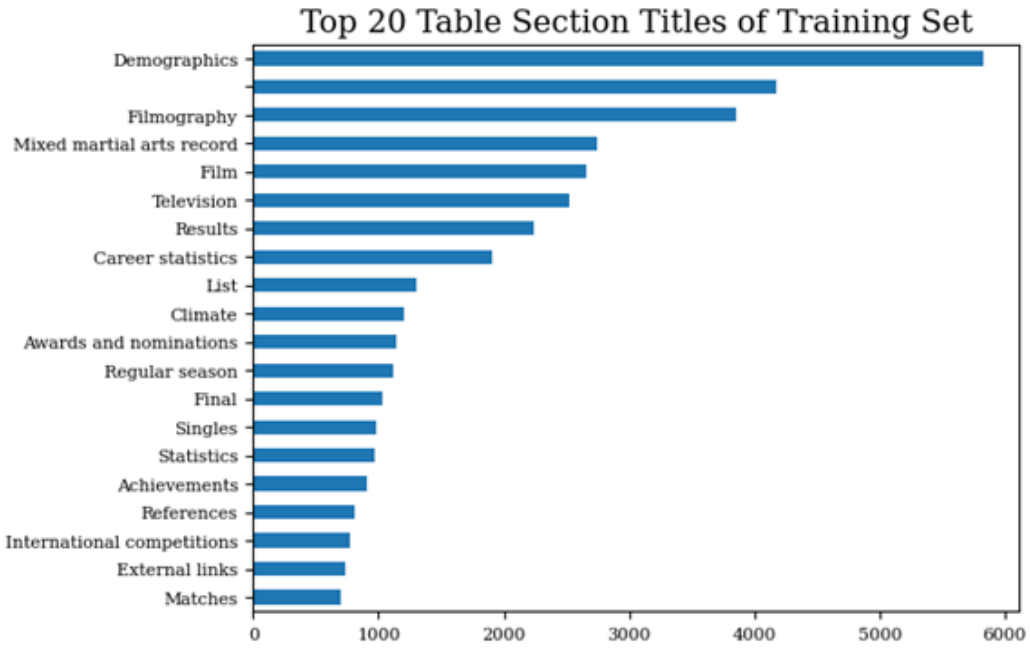
Next, the input templates are cleaned from any HTML tags and redundant punctuation marks. If a word appears more than once consecutively, only a singular instance is retained. Finally, the hash symbol is substituted with the word "number".

Due to the fact that the materialization of the training set templates requires a significant amount of time as it consists of more than 100K instances, a decision has been made to extract a stratified sample [Kha+22]. Stratified sampling ensures that the percentage of tables per subject remains the same even though the total number of examples is reduced. In total 23,650 tables are used which is the 25% of the previous training set.

Finally, it is important to mention that no text normalization is applied to the templates.



(a) A barplot illustrating the top 20 most frequent Table Section Titles of the stratified sample.



(b) A barplot illustrating the top 20 most frequent Table Section Titles of the training set before sampling.

Fig. 4.6: Barplots that show that the proportion of Table Section Titles of the Training Set matches the one of the Stratified Sample.

4.2 QTSumm

This dataset does not require much preprocessing as the tables do not appear to have any missing cells. Similar to the approach followed regarding ToTTo, as the models receive a sequence input, it is necessary to convert tables into a text format. Once again, the employed method involves creating templates where the table information is presented in a "Header is Value" arrangement. Next, the highlighted information is added, the number of table columns and rows along with the query.

The templates of the QTSumm dataset look like this:

Template = Query + Table Converted to Text + Focus on + Table Structure

Which competitors had their season's best (SB) performance in this championship and how did their results compare to each other? Table converted to text: Rank is 1, Name is Brianne Theisen-Eaton, Nationality is Canada, Time is 2:09.99, Points is 965, Notes is SB, Rank is 2, Name is Barbara Nwaba, Nationality is United States, Time is 2:10.07, Points is 964, Notes is SB, Rank is 3, Name is Györgyi Zsivoczky-Farkas, Nationality is Hungary, Time is 2:18.48, Points is 844, Notes is SB, Rank is 4, Name is Makeba Alcide, Nationality is Saint Lucia, Time is 2:18.65, Points is 842, Notes is SB, Rank is 5, Name is Salcia Slack, Nationality is Jamaica, Time is 2:19.00, Points is 837, Notes is , Rank is 6, Name is Kateřina Cachová, Nationality is Czech Republic, Time is 2:19.97, Points is 824, Notes is , Rank is 7, Name is Georgia Ellenwood, Nationality is Canada, Time is 2:20.18, Points is 821, Notes is , Rank is 8, Name is Morgan Lake, Nationality is Great Britain, Time is 2:20.40, Points is 818, Notes is , Rank is 9, Name is Alina Fyodorova, Nationality is Ukraine, Time is 2:20.42, Points is 818, Notes is , Rank is 10, Name is Kendell Williams, Nationality is United States, Time is 2:22.82, Points is 786, Notes is , Rank is 11, Name is Anastasiya Mokhnyuk, Nationality is Ukraine, Time is 2:23.19, Points is 781, Notes is , Rank is 12, Name is Celina Leffler, Nationality is Germany, Time is 2:24.01, Points is 770, Notes is , Focus on: 1, Brianne Theisen-Eaton, Canada, 2:09.99, 965, SB, 2, Barbara Nwaba, United States, 2:10.07, 964, SB, 3, Györgyi Zsivoczky-Farkas, Hungary, 2:18.48, 844, SB, 4, Makeba Alcide, Saint Lucia, 2:18.65, 842, SB, Table Structure: Number of rows: 12, Number of columns: 6

Fig. 4.7: An example of a table template - QTSumm dataset.

It is important to mention that no text regularization or cleaning is applied before passing the input data to the models.

Evaluation Metrics

5

5.1 ROUGE

Introduced by Chin-Yew Lin in 2004 [Lin04], ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It consists of a set of metrics used for the automatic evaluation of machine-generated text, particularly in the context of text summarization. ROUGE measures the quality of a generated summary by comparing it with reference summaries created by humans. The metrics focus on assessing the overlap and matching of n-grams, word sequences, and word pairs between the generated summary and the reference summaries. ROUGE ranges from 0 to 1, with higher values indicating a better match.

In this study, the following ROUGE metrics were used:

- ROUGE-N
- ROUGE-L
- ROUGE-Lsum

Given its widespread usage and standardized nature, ROUGE allows fair comparisons between different summarization systems. However, it only counts the number of n-grams and does not take the coherence and readability of the summaries under consideration.

5.1.1 ROUGE-N

This metric assesses the alignment of 'N-grams' between the summary produced by the model and a reference summary. For instance, consider a scenario where we have a Reference Summary labeled as R and a Candidate Summary denoted as C.

- R : "I love coffee with crushed ice."
- C : "I love iced coffee with whipped cream."

To calculate ROUGE-N, Precision and Recall must be computed. Precision is the number of words in C that also appear in R whereas Recall is the opposite, defined as the number of words in R that also appear in C. Finally, we compute the harmonic mean of precision and recall (F1-Score) using the formula:

$$F_1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.1}$$

For ROUGE-1 the words in C that also appear in R are: “I”, “love”, “coffee”, “with”. So, 4 out of 7. The words from R that also appear in C are the same. So, 4 out of 6. For ROUGE-2 the bigrams from C that also appear in R are: “I love” and “coffee with”. So, it’s 2 out of 6 and 2 out of 5. The table below presents the results in a compact form.

Metrics	ROUGE-1	ROUGE-2
Precision	4/7	2/6
Recall	4/6	2/5
F1-score	8/13	4/11

Tab. 5.1: Table including the results of calculations regarding Rouge-N

5.1.2 ROUGE-L

The Longest Common Subsequence (LCS) method identifies the longest sequence of words that manifests in the same order in both the reference and machine-generated summaries. ROUGE-L considers the summary as a whole ignoring where the lines break. Consider the following Reference (R) and Candidate (C) summaries :

- R : “I love coffee with crushed ice. It is delicious.”
- C : “I love iced coffee with whipped cream. It is so tasty.”

To calculate ROUGE-L, Precision and Recall are required as before. The LCS is “I love coffee with. It is” which is comprised of 6 words.

Consequently,it follows that Precision = 6/11 , Recall = 6/9 and F1-Score = 0,6 .

5.1.3 ROUGE-Lsum

ROUGE-Lsum is a variant of the ROUGE-L metric specifically designed to evaluate summaries at the sentence level. Instead of treating the entire text as a single sequence, ROUGE-Lsum first breaks down the summaries into individual sentences, calculates the ROUGE-L score for each sentence independently and then takes the average score for all sentences.

Considering the previous example of Reference and Candidate Summaries, the LCS for the first sentence is “I love coffee with” which consists of 4 words. So, the Precision in this case is equal to 4/7 and the Recall is equal to 4/6 which means that the F1-Score is 0,62. For the second sentence the LCS is “It is” which contains 2 words. Consequently, the Precision is 2/4 and the Recall is 2/3 which give a F1-Score of 0,56 .

In this case the ROUGE-Lsum is 0,59 .

5.2 SACREBLEU

The SacreBLEU metric calculates the BLEU (Bilingual Evaluation Understudy) score for a translated text proposed by Kishore Papineni et al.in 2002 [Pap+02]. BLEU measures the similarity between the machine-translated text and the reference translations. BLEU score can be computed using the formula below:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{5.2}$$

where BP is defined as:

$$\text{BP} = \begin{cases} 1, & \text{if } c > r \\ e^{(1-\frac{r}{c})}, & \text{if } c \leq r \end{cases}$$

In this formula:

- BP represents the Brevity Penalty, a penalty term that adjusts the score for translations that are shorter than the reference text.
- c is the length of the candidate translation.
- r is the effective reference corpus length.
- N is the number of grams considered .
- w_n is the weight for each gram.
- p_n is the precision for n-grams.

BLEU score ranges from 0 to 1. Values close to 1 indicate better translation quality with a perfect translation having a BLEU score of 1. A completely incorrect translation would have a BLEU score of 0. According to Ketan Doshi [Dos21], a score of 0,6 or 0,7 is considered the best possible in practice. This is because even two humans can generate different sentence variants, making the likelihood of achieving a perfect match highly improbable.

To ease the reader's understanding, an arithmetic example follows. Consider again the Reference Summary - Candidate Summary pair mentioned in 5.1.1. The pair is now adjusted to:

- R: "I love coffee with crushed ice."
- C: "I love coffee with whipped cream and ice."

To begin with, it is necessary to compute the precision scores for unigrams through fourgrams as typically the N of choice is 4. The unigram precision would be 5/8 (p1), the bigram precision (p2) would be 3/7 , the trigram precision (p3) would be 2/6 and the fourgram (p4) precision would be 1/5.

Next, the precision scores are combined using the second part of the formula 5.2. This can be computed for different values of N and with different weight values. For N = 4 and $w_n = N/4$ (uniform weights) the Geometric Average Precision (GAP) (eq.5.3) would be:

$$\begin{aligned}\text{GAP (N)} &= \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \\ &= \exp(w_1 \cdot \log p_1 + w_2 \cdot \log p_2 + w_3 \cdot \log p_3 + w_4 \cdot \log p_4) \\ &= exp(1/4 \cdot \log(5/8) + 1/4 \cdot \log(3/7) + 1/4 \cdot \log(2/6) + 1/4 \cdot \log(1/5)) \\ &\approx exp(0.25 \cdot (-0.20) + 0.25 \cdot (-0.37) + 0.25 \cdot (-0.48) + 0.25 \cdot (-0.7)) \\ &\approx exp(-0.44) \approx 0.64\end{aligned} \tag{5.3}$$

Since the number of words in the predicted sentence (C) is 8 and the number of words in the reference sentence (R) is 6, the BP is equal to 1.

So, the BLEU score in this example is :

$$\text{BLEU} = \text{BP} \cdot \text{GAP} = 1 \cdot 0.64 = 0.64 \tag{5.4}$$

Although BLEU is a known and trusted metric, Matt Post [Pos18] claims that there are some issues when it comes to reporting its scores, namely:

- BLEU is not a single metric but requires a number of parameters.
- preprocessing schemes have a large effect on scores. Importantly, BLEU scores computed against differently-processed references are not comparable.
- papers vary in the hidden parameters and schemes they use, yet often do not report them. Even when they do, it can be hard to discover the details.

In the paper "A Call for Clarity in Reporting BLEU Scores", published in 2018, SACREBLEU [Pos18] is proposed to tackle those issues.

5.3 METEOR

METEOR, a metric proposed by Satanjeev Banerjee et al. in 2005 [BL05], stands for Metric for Evaluation of Translation with Explicit Ordering. It serves as a measure for assessing machine translation by comparing it with human translations. Unlike other metrics, it takes into account the order in which words appear in the translation. METEOR considers precision, fluency, and the sequential arrangement of words in the evaluation. The score falls within the range of 0 to 1, with a higher score signifying superior translation quality.

METEOR evaluates a machine translated text against a human reference translation by segmenting them into chunks. A chunk is a set of consecutive words. The similarity between each chunk is assessed by using metrics like unigram precision, recall, and F-score, bigram overlap, and exact word matches. Finally, the METEOR score is determined by the weighted average of these measures.

Consider the following Candidate Text - Reference Text pair:

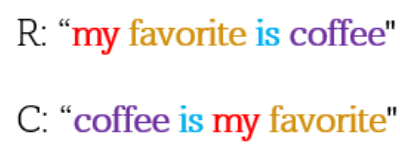


Fig. 5.1: Example of word for word alignment.

Aligning the generated text with the annotation can be achieved either through a word-for-word matching approach or by employing tools for similarity, such as word embeddings and dictionaries, among other methods. In Fig. 5.1 common words found in the Reference and Candidate text are highlighted using different colors.

While there are multiple alignments possible, the aim is to pick the alignment with the lowest number of chunks.

For instance, consider another two sentences:

- R: “the boss loves the coffee”
- C: “the boss adores the coffee”

In this example, the word “the” in the candidate translation can be mapped to either of the two “the”s in the reference. Mapping it to the first appears to be a better approach as it leads to just one chunk. This is due to the fact that all words in the candidate translation map to the reference one consecutively: the – the, boss – boss, loves – adores, the – the, coffee - coffee.

To calculate the METEOR score, it is necessary to compute precision and recall. Considering the example of Fig. 5.1, both the precision and recall are equal to 1 since all the words in the candidate text appear in the reference text and they both have the same number of unigrams. As a result, the F score (equation 5.5) is also equal to 1.

$$F_{mean} = \frac{10PR}{R + 9P} \tag{5.5}$$

Next, the Chunk Penalty (p) is calculated using the formula provided in equation 5.6. This is a penalty based on the number of chunks in the candidate text that map to chunks in the reference text. Ideally, if the candidate and reference texts are identical, all words in the candidate map to the words in the reference consecutively, resulting in a single chunk.

$$\mathrm{p} = 0.5 \cdot \left(\frac{c}{u_m}\right)^3 \tag{5.6}$$

In formula 5.6, the u_m stands for the number of unigrams in the candidate text and the c stands for the number of chunks in the candidate text.

In the example shown in Fig. 5.1, there are three chunks: "my favorite", "is", and "coffee". As a result, c equals 3. The candidate text consists of four unigrams: "coffee", "is", "my", "favorite". Thus, u_m equals 4. So, the Chunk Penalty is approximately equal to 0.42 .

Finally, the METEOR score is calculated by using the formula 5.7 and combining all the above:

$$\begin{aligned} \mathrm{M} &= F_{mean} \cdot (1 - p) \\ &= 1 \cdot (1 - 0.42) \\ &= 0.58 \end{aligned} \tag{5.7}$$

6 Models

The guided summaries are produced by using 3 different transformer models, namely T5-small, T-base and Bart-base. The models are fine-tuned on ToTTo and QTSumm without freezing any layers. In order to make sure that the performance of the models is the most optimal, hyperparameter tuning is implemented regarding batch size and learning rate as in most cases these two parameters affect the performance the most. Additionally, to prevent overfitting, decoupled weight decay regularization is utilized [LH17].

The term batch size refers to the number of training examples used in one iteration whereas learning rate determines the size of the steps taken during the optimization process, controlling how quickly or slowly a model learns. The learning rate is a scalar value that multiplies the gradient, which is the partial derivative of the loss with respect to the model parameters.

6.1 T5

Introduced by Google Research in 2019 [Raf+19], T5 treats various natural language processing tasks such as translation, summarization and question-answering. The model utilizes an encoder-decoder setup where it employs stacked multi-head self-attention and position-wise feed-forward layers, complemented by residual connections and layer normalization at each stage. T5 introduces positional information by incorporating sinusoidal positional encoding into the input embeddings before processing. The self-attention mechanism, utilizing multiple heads of attention, empowers the model to simultaneously focus on various elements within the input sequence, facilitating the learning of diverse contextual relationships. During the pre-training phase, T5 adopts a denoising autoencoder setup with a masked language modeling objective. This involves randomly masking input tokens and training the model to predict them from unmasked tokens. In this project the smallest version (60M parameters) and the base version (220M) were used. The smaller version has 6 layers in the encoder and decoder along with 8 attention heads. Regarding the base version, both the encoder and decoder consist of 12 blocks and 12 attention heads.

6.2 Bart

Developed by the AI team of Facebook in 2019 [Lew+19], the Bidirectional and Auto-Regressive Transformers (BART), is a model utilized for many NLP tasks such as summarization, translation, and text generation. It uses a standard sequence-to-sequence Transformer architecture with a bidirectional encoder (like BERT [Dev+18]) and a left-to-right decoder (like GPT [Yen+23]). In the preprocessing phase, some tokens of the input are initially masked or deleted to create an incomplete, altered version of the original sequence. Next, the denoising part follows where BART learns to reconstruct the original sentence from the incomplete version. Through this process, the model gains insights into the structure and semantics of the input it processes. Through an encoder-decoder framework, the encoder maps the altered input to a latent representation and the decoder produces the original sentence based on the representation generated by the encoder. In this project the base version (140M parameters) with 6 layers in the encoder and decoder is employed. Both the encoder and the decoder have 12 attention heads.

7 Results

In this section the training parameters as well as the results of all the conducted experiments are presented. In order to get the ROUGE, METEOR and SACREBLEU scores for the whole test set, the mean, median and standard deviation are computed after calculating the scores for each generated summary and annotation pair.

7.1 ToTTo

While using the ToTTo dataset T5-small and Bart-base are trained for 4 epochs each while T5-base is trained for 1 epoch. The training is performed using all data (23,650 examples for training and 12,076 examples for validation). In order to specify the task, the instruction "summarize:" is added to the beginning of each example for all sets.

Next, the modified templates and the annotations are tokenized with a maximum length of 1,024 tokens and 128 tokens accordingly. If the input exceeds the specified maximum length, in both cases, it gets truncated.

After training the models, evaluation is performed first for the whole test set (7,700 samples) and then per domain. The top 5 most popular domains are chosen. The categories of "Film", "Filmography" and "Television" are merged into one which is named "Film/Tv". The tables that do not have a domain label are denoted as "Unspecified". Tab. 7.1 presents the number of examples per domain:

Domain	Number of Examples
Film/TV	600
Demographics	417
Unspecified	336
Mixed Martial Arts Record	158
Career statistics	152

Tab. 7.1: Number of examples of the Top 5 most popular domains of the ToTTo dataset.

7.1.1 T5-small

For the T5-small, after performing hyperparameter tuning, a batch size of 4 is selected. After taking into account the available resources, the suggested values were 4, 8 and 16. Regarding learning rate, the given values were 2e-5, 2e-6 and 2e-4, again considering the resources at hand. It came out that the best option is 2e-5. Moreover, a choice is made to opt for a weight decay of 10^{-2} to prevent overfitting by discouraging the model from assigning excessively large weights to its parameters. The training process lasts about 2 hours on a T4 GPU.

Fig. 7.1 provides the training and validation loss curves. Both losses decrease by the passage of the epochs which is an indication that the model learns from the training data and is able to leverage this knowledge to make predictions on the validation set. However, it is noteworthy that the validation loss curve is not significantly steep.

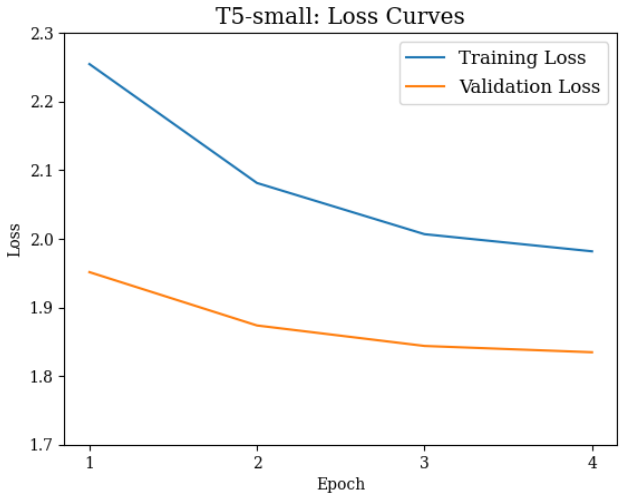


Fig. 7.1: Loss curves of T5-small trained on ToTTo dataset.

Moving on to the results, as highlighted in Fig. 7.2, the model demonstrates satisfactory performance across all categories. Amongst all domains, it seems to perform the best in "MMA Record" (Mixed Martial Arts) category followed by "Demographics" and "Film/Tv". The weakest performance is observed in "Career Statistics".

To be more specific, regarding Rouge-1, the model performs relatively well across all domains, with higher scores in categories such as "Film/TV" and "MMA Record", indicating that there is a significant amount of overlap between the unigrams of the machine-generated summaries and the human annotations.

By looking at Rouge-2 it is evident that the model demonstrates better performance in the "MMA Record" category with lower scores being present in "Career Statistics".

To continue, the scores of Rouge-L and Rouge-Lsum denote that there is a decent amount of overlap between the longest common subsequence of words present in the machine-generated text and the human-generated annotation.

The score of SacreBleu shows a significant variation across domains, with high scores again noticed in “MMA Record” group.

The Meteor score also varies across domains, with the highest score observed in “MMA Record”.

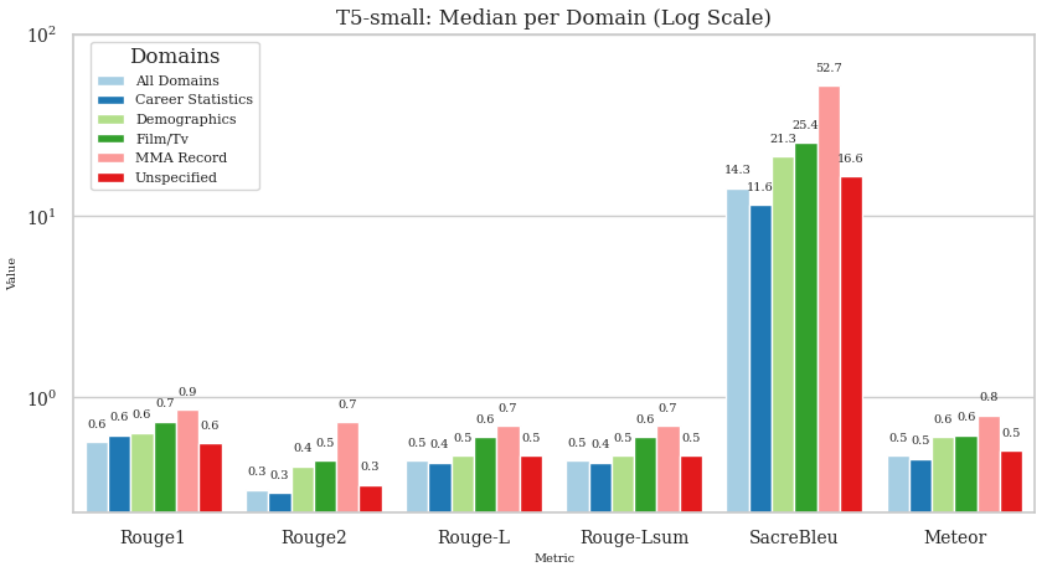


Fig. 7.2: Median of T5-small evaluation metrics per domain - Totto dataset.

To get a well rounded view of the models’ performance, below are provided three tables including the mean, median and standard deviation along with their corresponding violin plots. Tab. 7.4 it shows that "Demographics" have the highest standard deviation across all metrics. Tab. 7.2 and Tab. 7.3 highlight the model’s capability to generate targeted text summaries from tables that derive from the "MMA Record" domain as in this category appear the highest scores.

Tab. 7.2: T5-small Results on ToTTo: Average

			Average			
Metric	**All**	**Career Statistics**	**Demographics**	**Film/Tv**	**MMA Record**	**Unspecified**
Rouge1	0.57	0.58	0.70	0.72	**0.84**	0.56
Rouge2	0.34	0.29	0.49	0.46	**0.70**	0.37
Rouge-L	0.48	0.44	0.59	0.62	**0.71**	0.51
Rouge-Lsum	0.48	0.44	0.59	0.62	**0.71**	0.51
SacreBleu	21.00	14.89	37.25	29.81	**54.75**	24.62
Meteor	0.49	0.45	0.63	0.62	**0.78**	0.51

Tab. 7.3: T5-small Results on ToTTo: Median

Median						
Metric	**All**	**Career Statistics**	**Demographics**	**Film/Tv**	**MMA Record**	**Unspecified**
Rouge1	0.57	0.62	0.64	0.74	**0.86**	0.56
Rouge2	0.31	0.30	0.42	0.45	**0.73**	0.33
Rouge-L	0.45	0.44	0.48	0.61	**0.70**	0.48
Rouge-Lsum	0.45	0.44	0.48	0.61	**0.70**	0.48
SacreBleu	14.29	11.57	21.34	25.37	**52.71**	16.59
Meteor	0.48	0.46	0.61	0.62	**0.80**	0.51

Tab. 7.4: T5-small Results on ToTTo:Standard Deviation

Standard Deviation						
Metric	**All**	**Career Statistics**	**Demographics**	**Film/Tv**	**MMA Record**	**Unspecified**
Rouge1	0.20	0.18	**0.22**	0.14	0.15	**0.22**
Rouge2	0.23	0.18	**0.30**	0.19	0.22	0.26
Rouge-L	0.21	0.15	**0.28**	0.17	0.21	0.23
Rouge-Lsum	0.21	0.15	**0.28**	0.17	0.21	0.23
SacreBleu	19.56	11.50	**30.96**	19.83	24.99	22.72
Meteor	0.22	0.19	**0.25**	0.19	0.18	0.24

Fig. 7.3 gives an insight regarding how the SACREBLEU scores are distributed. It appears that the majority of the curves are positively skewed which means that it's more probable to get smaller scores, especially in "All" and "Career Statistics" categories.

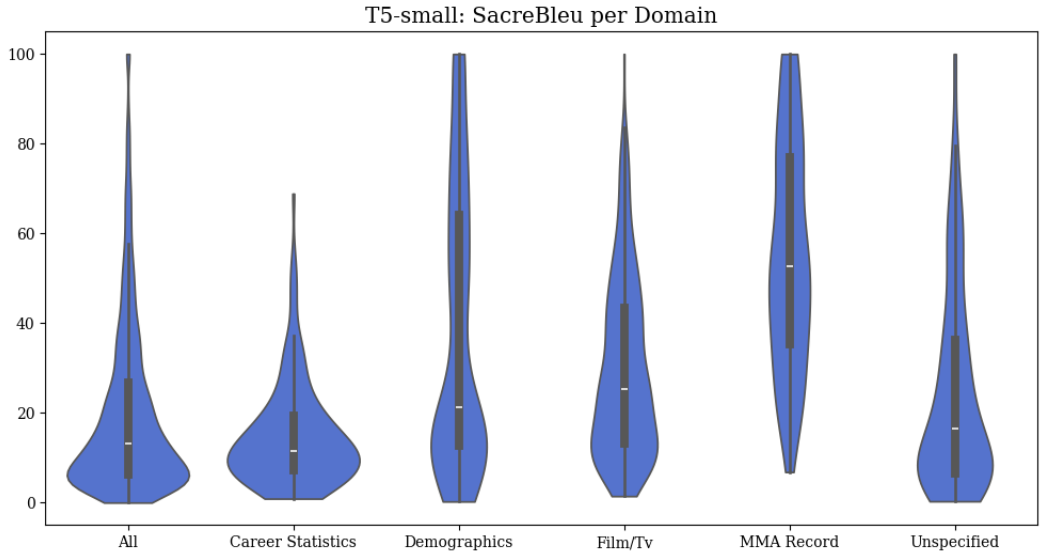


Fig. 7.3: Violin Plot: SACREBLEU scores of T5-small fine-tuned on ToTTo per domain.

Regarding METEOR, as Fig. 7.4 shows, the distribution of "Demographics" is bimodal with the most probable values appearing around 0.5 and 0.9. The curve of "MMA Record" it's negatively skewed. This means that it's more probable to get higher METEOR scores in this domain.

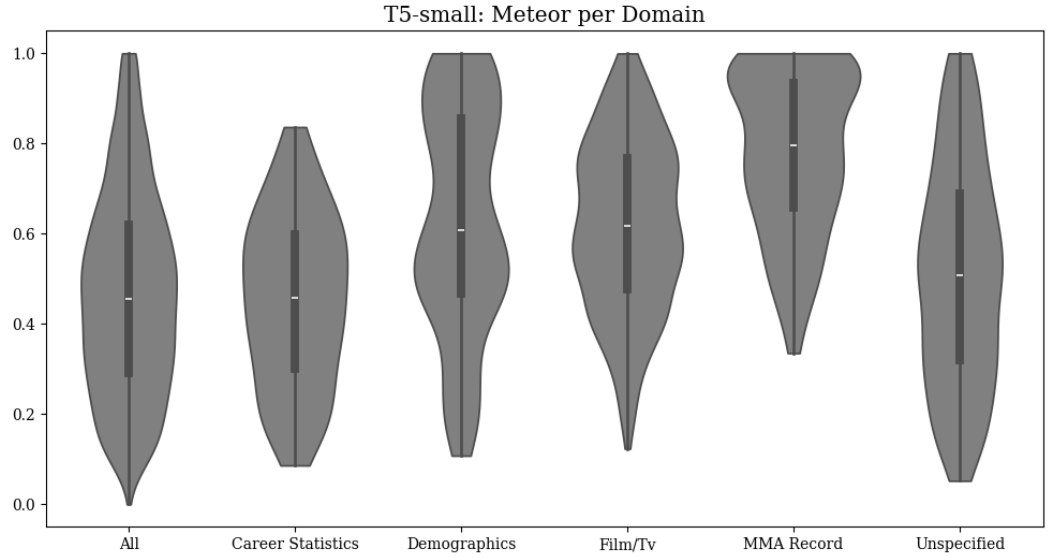


Fig. 7.4: Violin Plot: METEOR scores of T5-small fine-tuned on ToTTo per domain.

The curves of ROUGE 1, ROUGE 2, ROUGE L and ROUGE Lsum of "Demographics", as portrayed in Fig. 7.5, seem to adopt almost the same behavior as METEOR having two peaks.

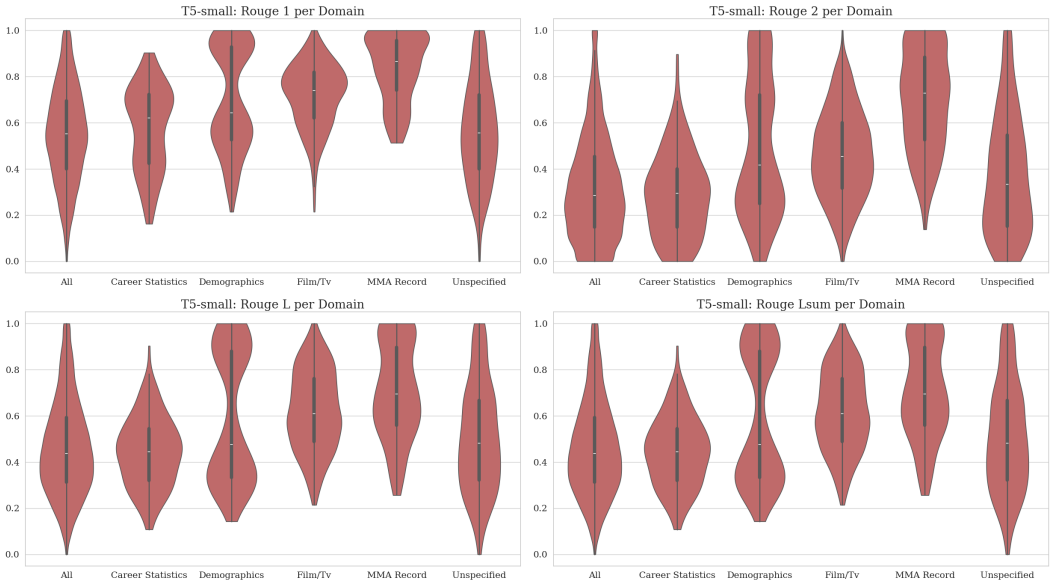


Fig. 7.5: Violin Plot: ROUGE scores of T5-small fine-tuned on ToTTo per domain.

The inference phase for the whole test set lasts around 30 minutes whereas for each individual category the maximum time needed is around 2 minutes. Tab.7.5 contains a sample of machine-generated summaries along with their corresponding annotations. The minimum length is set to 5 whereas the maximum length is set to 30.

Tab. 7.5: T5-small: Generated Summary VS Annotation - ToTTo dataset

Table	Generated Summary	Annotation
0	Daniel Henry Chamberlain served as the Governor of South Carolina from December 1, 1874.	Daniel Henry Chamberlin was the 76th Governor of South Carolina from 1874.
1	In 2016, Jodorowsky played Evelyn in Kids in Love.	Alma Jodorowsky had the role of Evelyn in the 2016 film Kids in Love.
2	In 2006, A. J. Hawk scored 119 touchdowns in 119 games.	In his rookie season, Hawk led with 119 total tackles.

To check whether the provided information is valid it is necessary to visit the Wikipedia page of the corresponding table. Regarding Table 0, according Fig. 7.6, Daniel Henry Chamberlain was indeed elected in December 1, 1874 as the governor of South Carolina but in the generated summary it is not mentioned that he was the 76th Governor of the area.

75		Franklin J. Moses Jr. (1838–1906) [136][137]	December 3, 1872[138] – December 1, 1874 (lost nomination)	Republican[9]	1872	Richard Howell Gleaves
76		Daniel Henry Chamberlain (1835–1907) [139][140]	December 1, 1874[141] – April 11, 1877 (lost election)	Republican[9]	1874 1876[n]	
77		Wade Hampton III (1818–1902) [142][143]	December 14, 1876[144] – February 26, 1879 (resigned)[o]	Democratic[9]	1878	William Dunlap Simpson

Fig. 7.6: A snippet of a Wikipedia table obtained from the page "List of governors of South Carolina".

The generated summary of Table 1 includes exactly the requested information which is given in a coherent sentence. Table 2 seems to confuse the model. The value included in the highlighted cell is “119” and it refers to the total tackles of A. J. Hawk in 2006 (when he was a rookie). The generated summary states that he scored 119 touchdowns in 119 games which is not valid according to the corresponding Wikipedia table (Tab.7.7).

Year	Team	GP	Tackles				Fumbles		Interceptions					
			Cmb	Solo	Ast	Sck	FF	FR	Int	Yds	Avg	Lng	TD	PD
2006	GB	**16**	119	**82**	37	3.5	**1**	**2**	2	31	15.5	25	0	8
2007	GB	**16**	105	78	27	1.0	**1**	1	1	10	10.0	10	0	4
2008	GB	**16**	86	67	19	3.0	0	0	0	0	0.0	0	0	1
2009	GB	**16**	89	67	22	1.0	**1**	0	2	**42**	**21.0**	**29**	0	2
2010	GB	**16**	111	72	39	0.5	0	1	**3**	31	10.3	21	0	**10**
2011	GB	14	84	53	31	1.5	0	0	0	0	0.0	0	0	3
2012	GB	**16**	**120**	81	39	3.0	0	0	0	0	0.0	0	0	0
2013	GB	**16**	118	74	**44**	**5.0**	**1**	1	1	7	7.0	7	0	4
2014	GB	**16**	89	53	36	0.5	0	0	0	0	0.0	0	0	2
2015	CIN	**16**	24	16	8	1.0	0	0	0	0	0.0	0	0	0
2016	ATL	1	0	0	0	0.0	0	0	0	0	0.0	0	0	0
Total[41]		**159**	**947**	**644**	**303**	**20.0**	**4**	**5**	**9**	**121**	**13.4**	**29**	**0**	**35**

Fig. 7.7: A Wikipedia table obtained from the page "A. J. Hawk".

7.1.2 Bart-base

Moving on to Bart-base, a batch size of 8 is selected. The chosen learning rate is 2e-5 and the weight decay is 10^{-2}. Even though the suggested batch size and learning rate from hyperparameter tuning are 16 (having 4, 8 and 16 as possible options) and 2e-6 (having 2e-6, 2e-5 and 2e-4 as possible options), the results are not as good as the original choice. So, it was empirically chosen to train the model setting the batch size to 8 and the learning rate to 2e-5. The training process lasts about one and a half hours on A100.

From Fig. 7.8 we notice that the validation loss smoothly decreases from 1.45 to around 1.38 whereas the training loss falls rapidly from a value close to 1.60 to a value close to 1.25 in the 3rd epoch. Then it increases again to 1.30 .

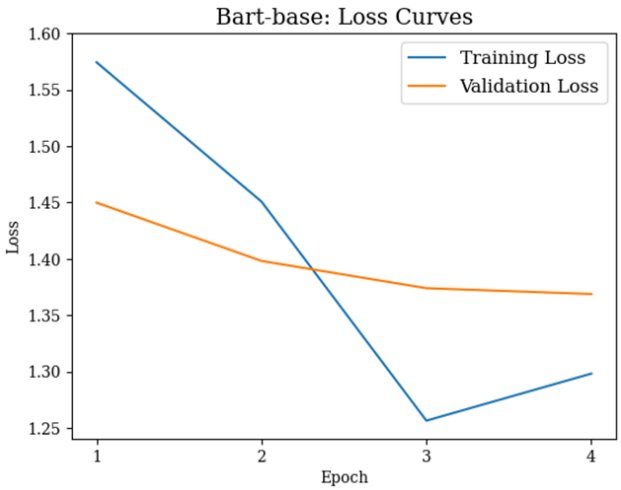


Fig. 7.8: Loss curves of Bart-base fine-tuned on ToTTo.

Proceeding to the results, the model seems in general to perform moderately with some categories being an exception. Notably, there is a significant amount of overlap between the n-grams of generated translations and the provided annotations, as indicated by high Rouge1, Rouge2, Rouge-L, and Rouge-Lsum scores, in the "Demographics" and "Film/Tv" categories. Additionally, the "MMA Record" and "Career Statistics" domains showcase acceptable performance as indicated by in Rouge1 and Rouge2 metrics. The Rouge-L and Rouge-Lsum seem to be lower demonstrating that the generated text has a relatively shorter common subsequence with the reference text. The SacreBleu and Meteor metrics suggest high linguistic quality within the "Demographics" category.

It arises that the highest scores across all metrics appear in "Demographics" according to Tab. 7.6 and Tab. 7.7 but it is important to mention that this category also has the highest standard deviation, an insight obtained from 7.8.

Tab. 7.6: Bart-base Results on ToTTo: Average

Average

Metric	All	Career Statistics	Demographics	Film/Tv	MMA Record	Unspecified
Rouge1	0.56	0.55	**0.78**	0.69	0.53	0.46
Rouge2	0.35	0.30	**0.58**	0.45	0.38	0.30
Rouge-L	0.47	0.44	**0.69**	0.60	0.43	0.37
Rouge-Lsum	0.48	0.46	**0.69**	0.61	0.44	0.41
SacreBleu	22.50	16.78	**47.65**	29.49	26.43	17.45
Meteor	0.53	0.51	**0.69**	0.64	0.59	0.53

Tab. 7.7: Bart-base Results on ToTTo: Median

Median

Metric	All	Career Statistics	Demographics	Film/Tv	MMA Record	Unspecified
Rouge1	0.56	0.53	**0.87**	0.71	0.48	0.46
Rouge2	0.31	0.26	**0.67**	0.44	0.32	0.26
Rouge-L	0.44	0.40	**0.80**	0.59	0.37	0.33
Rouge-Lsum	0.46	0.42	**0.80**	0.60	0.37	0.39
SacreBleu	14.57	11.20	**52.04**	24.66	17.85	11.50
Meteor	0.53	0.51	**0.78**	0.64	0.62	0.54

Tab. 7.8: Bart-base Results on ToTTo: Standard Deviation

Standard Deviation

Metric	All	Career Statistics	Demographics	Film/Tv	MMA Record	Unspecified
Rouge1	0.23	0.19	0.20	0.17	**0.26**	0.22
Rouge2	0.24	0.20	**0.33**	0.21	0.28	0.21
Rouge-L	0.23	0.18	**0.28**	0.19	0.22	0.22
Rouge-Lsum	0.23	0.18	**0.28**	0.18	0.22	0.22
SacreBleu	22.03	15.71	**33.55**	20.66	26.43	17.29
Meteor	0.22	0.19	**0.26**	0.18	**0.26**	0.21

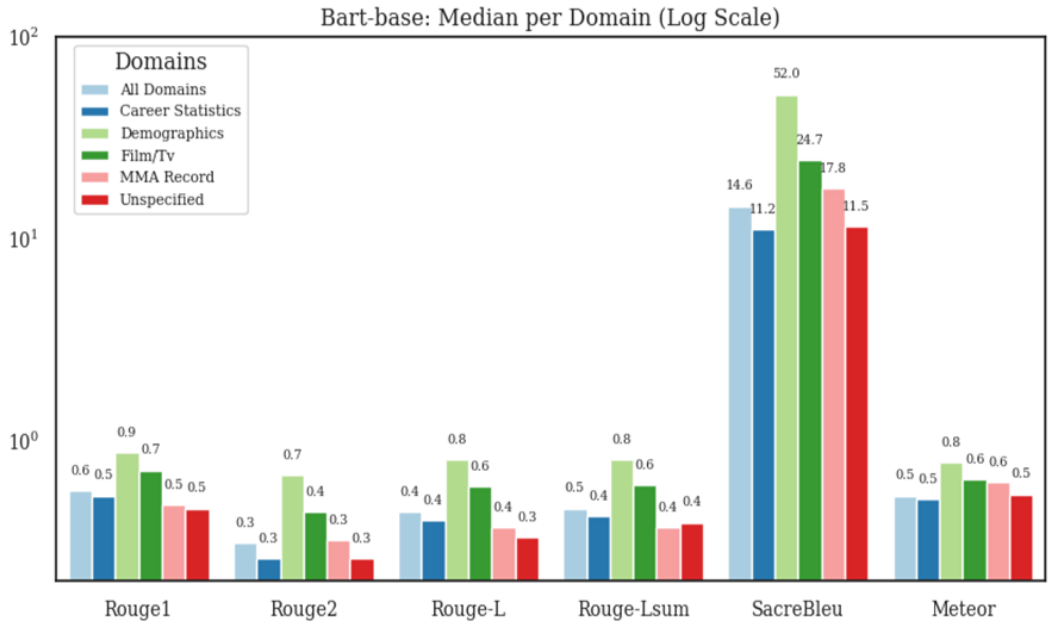


Fig. 7.9: Bar Plot: Median of Bart-base evaluation metrics per domain on ToTTo dataset

In Fig. 7.10, it appears that the majority of the distributions are positively skewed. This means that, in most domains, the scores tend to range towards the lower part of the SACREBLEU spectrum. The exception is the curve of "Demographics" as the values appear almost with the same frequency. The distribution of this category, according to Tab. 7.8, has the highest standard deviation across all domains, possibly justifying its wider shape.

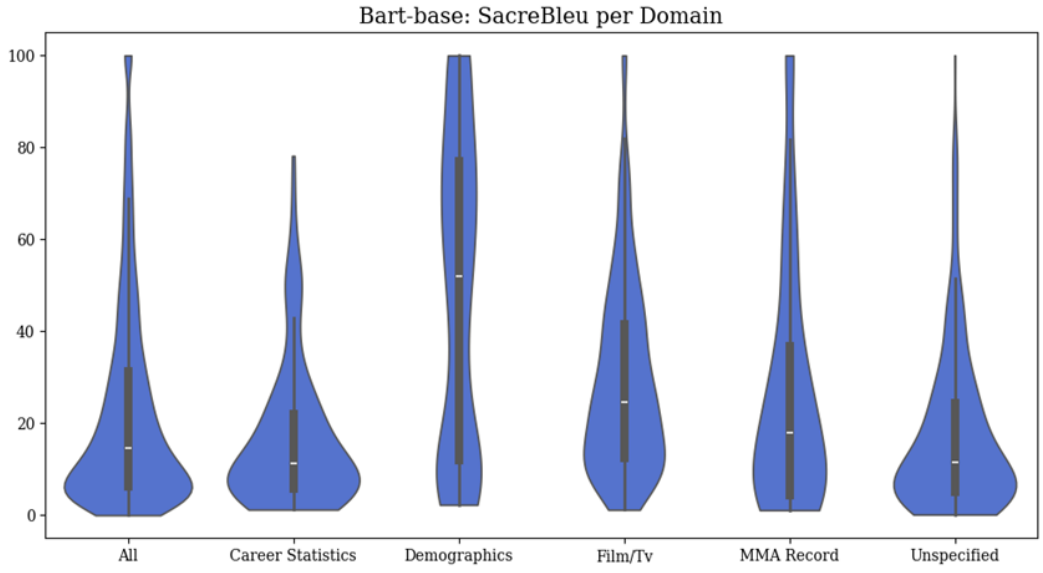


Fig. 7.10: Violin Plot: SACREBLEU scores of Bart-base fine-tuned on ToTTo per domain.

Regarding Meteor, Fig. 7.11 includes more symmetrical shapes with the exception being the violin plot of "Demographics". It seems that the most probable values in this case range between 0.8 and 1.

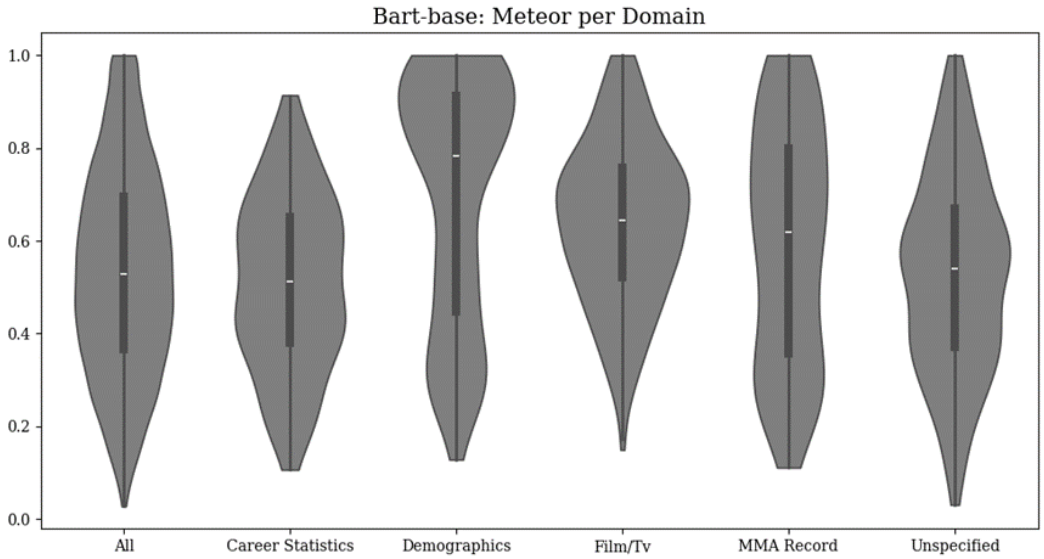


Fig. 7.11: Violin Plot: METEOR scores of Bart-base fine-tuned on ToTTo per domain.

An other interesting finding derives from Fig. 7.12. As one may notice in the plots of Rouge-1, Rouge-L and Rouge-Lsum for the categories of "Career Statistics" and "Demographics" the corresponding curves do not include any values below 0.15. The most eye-catching shape belongs to "Demographics" as the distribution is bimodal and negatively skewed denoting that a great portion of summaries regarding this domain is more probable to have higher ROUGE scores.

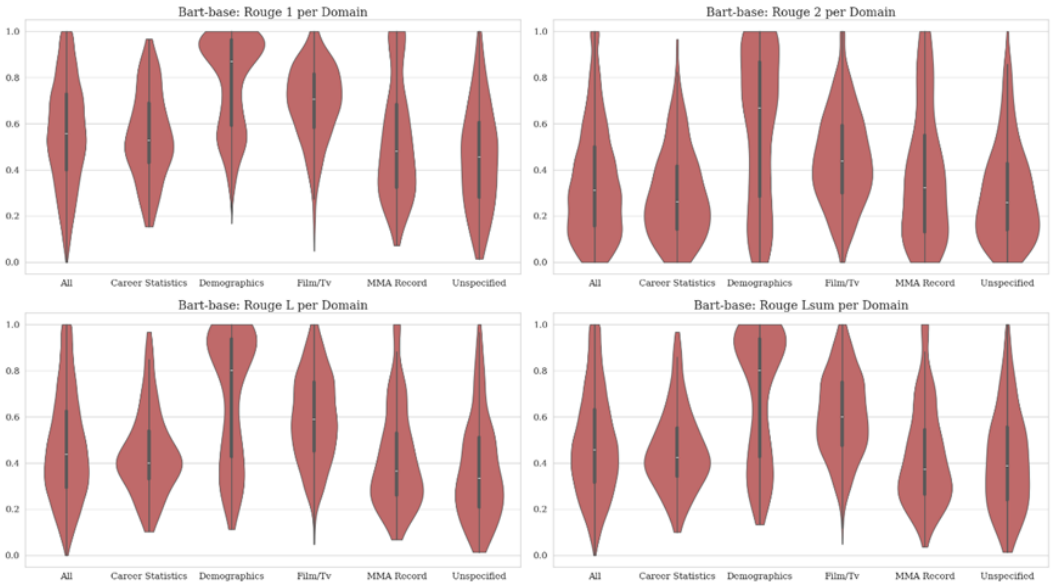


Fig. 7.12: Violin Plot: ROUGE scores of Bart-base fine-tuned on ToTTo per domain.

Next, in Tab. 7.9 a sample of machine-generated summaries along with their corresponding annotations is given. The minimum length is set to 5 and the maximum length is set to 30. For the whole test set the time needed for inference is around 1 hour whereas for each domain it is at maximum around 2 minutes.

Tab. 7.9: Bart-base: Generated Summary VS Annotation - ToTTo dataset

Table	Generated Summary	Annotation
0	Daniel Henry Chamberlain was the 76th Governor of South Carolina.	Daniel Henry Chamberlin was the 76th Governor of South Carolina from 1874.
1	Alma Jodorowsky played Evelyn in the 2016 Kids in Love.	Alma Jodorowsky had the role of Evelyn in the 2016 film Kids in Love.
2	A. J. Hawk finished the 2006 season with 119 tackles, nine sacks, and three forced fumbles.	In his rookie season, Hawk led with 119 total tackles.

Observations on Tab. 7.9:

- Table 0: It appears that almost all required information is included in the guided summary except for the fact that Daniel Henry Chamberlain governed from 1874. Notice that the two summaries are almost identical and the generated text does not appear to suffer from any grammatical or syntactical errors.
- Table 1: All required information is included in the generated text. The summary is coherent and contextually relevant.
- Table 2: In order to assess the generated summary, it is necessary to look at the corresponding table 7.7 as the model included more information that needed. Indeed in 2006 (when he was a rookie) Hawk had in total 119 tackles but the rest of the information included in the summary is not accurate.

7.1.3 T5-base

For the T5-base, a batch size of 4, a learning rate of 2e-5 and a weight decay of 10^{-1} are empirically chosen. The model was also trained using 2e-6 but the results did not improve, so 2e-5 is the final choice of learning rate. The training process lasts about 4 hours on a T4 GPU. Since the model is trained for only one epoch there are no loss curves available. Proceeding with the results:

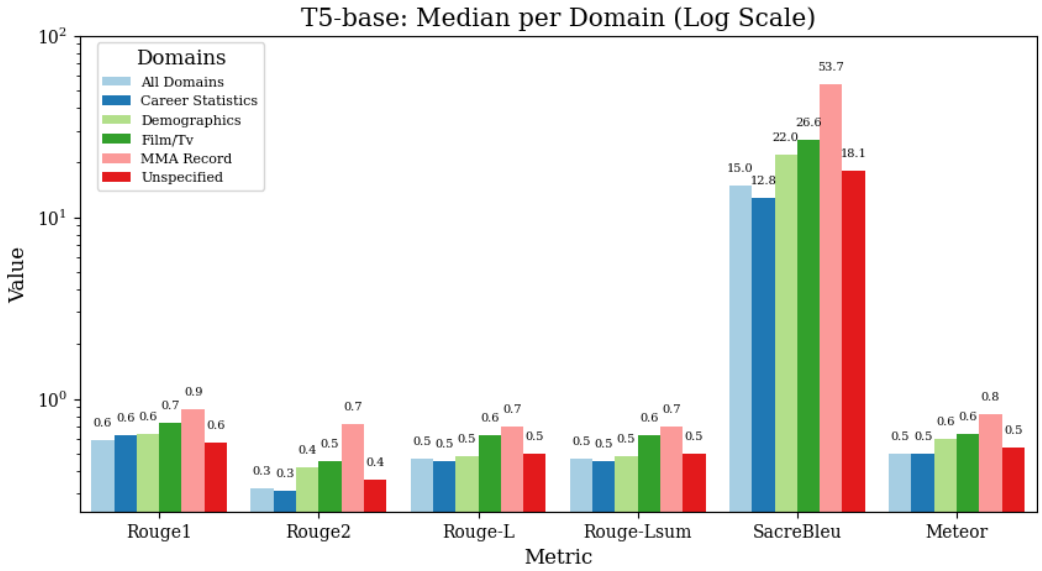


Fig. 7.13: Bar Plot: Median of T5-base evaluation metrics per domain on ToTTo dataset

According to Tab. 7.10 and Tab. 7.11, the model is very good at providing guided summaries for tables that belong to MMA Record domain. The Rouge metrics collectively indicate that the model generates text that aligns well with the reference summaries, capturing both individual words and longer sequences effectively. The SACREBLEU scores again highlight the model's strength in the "MMA Record" domain. Regarding the remaining categories, the performance is fair. According to Meteor scores, the performance could be considered moderate to high with the highest scores being present in the MMA Record category followed by Film/Tv and Demographics. The highest standard deviation across almost all metrics, according to Tab. 7.12, belongs to the category of "Demographics".

Tab. 7.10: T5-base Results on ToTTo: Average

			Average			
Metric	**All**	**Career Statistics**	**Demographics**	**Film/Tv**	**MMA Record**	**Unspecified**
Rouge1	0.58	0.60	0.70	0.72	**0.83**	0.58
Rouge2	0.35	0.32	0.49	0.46	**0.67**	0.40
Rouge-L	0.49	0.47	0.59	0.62	**0.67**	0.53
Rouge-Lsum	0.49	0.47	0.59	0.62	**0.67**	0.53
SacreBleu	22.05	16.81	37.66	30.69	**51.43**	26.13
Meteor	0.50	0.50	0.62	0.64	**0.77**	0.53

Tab. 7.11: T5-base Results on ToTTo: Median

Median						
Metric	**All**	**Career Statistics**	**Demographics**	**Film/Tv**	**MMA Record**	**Unspecified**
Rouge1	0.59	0.63	0.64	0.74	**0.87**	0.57
Rouge2	0.32	0.31	0.42	0.45	**0.73**	0.36
Rouge-L	0.47	0.45	0.48	0.63	**0.70**	0.50
Rouge-Lsum	0.47	0.45	0.48	0.63	**0.70**	0.50
SacreBleu	15.01	12.82	21.99	26.60	**53.75**	18.14
Meteor	0.50	0.50	0.60	0.64	**0.82**	0.54

Tab. 7.12: T5-base Results on ToTTo: Standard Deviation

Standard Deviation						
Metric	**All**	**Career Statistics**	**Demographics**	**Film/Tv**	**MMA Record**	**Unspecified**
Rouge1	0.21	0.17	0.22	0.13	0.17	**0.23**
Rouge2	0.23	0.19	**0.30**	0.19	0.22	0.26
Rouge-L	0.21	0.16	**0.29**	0.16	0.20	0.24
Rouge-Lsum	0.21	0.16	**0.29**	0.16	0.20	0.24
SacreBleu	20.19	13.46	**31.28**	19.43	24.13	22.83
Meteor	0.23	0.19	**0.26**	0.18	0.20	0.25

In Fig. 7.14 the same pattern appears with all distributions being positively skewed. For "Career Statistics" it seems that there are no SACREBLEU scores surpassing 60. In Fig. 7.15 it shows that only two categories, namely "All" and "Unspecified", have scores ranging from 0 to 1. The other categories have a smaller range with "Career Statistics" having the smallest. In this case the METEOR scores range from 0.1 to 0.9.

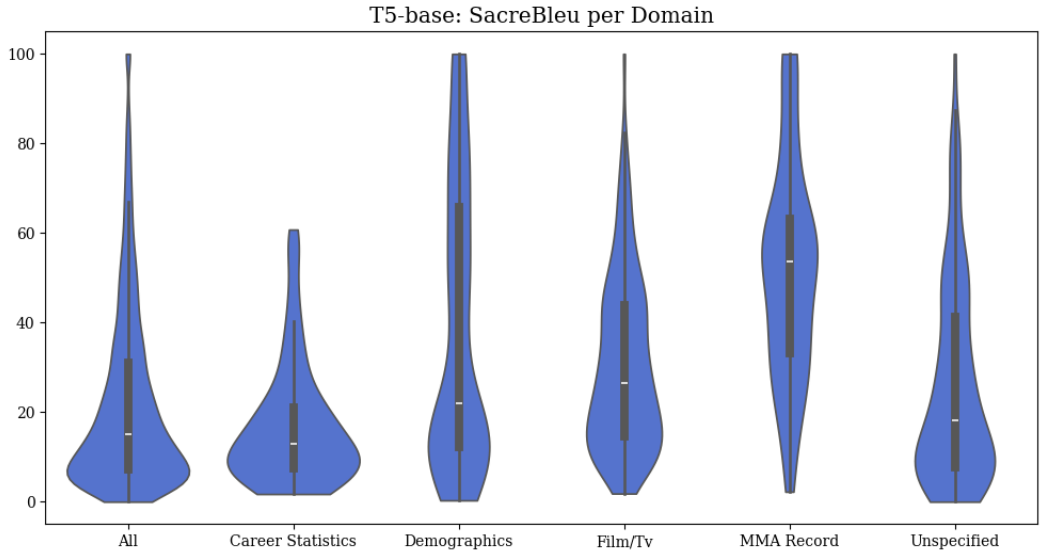


Fig. 7.14: Violin Plot: SACREBLEU scores of T5-base fine-tuned on ToTTo per domain.

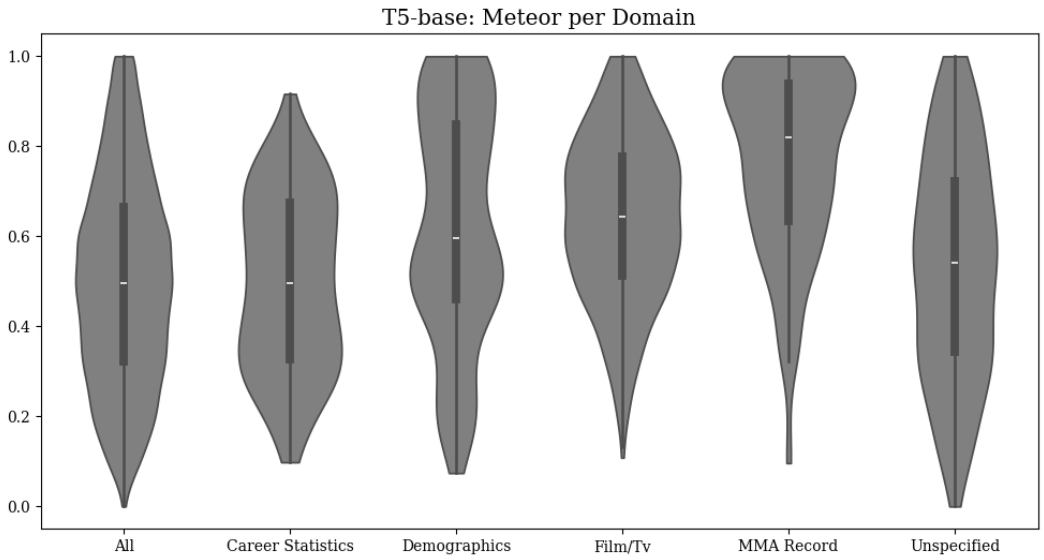


Fig. 7.15: Violin Plot: METEOR scores of T5-base fine-tuned on ToTTo per domain.

Moving on to Fig. 7.16 it arises that in ROUGE 1 plot for the domain of "Career Statistics" the values range from 0.2 to 0.9. The category with the widest violin is "Film/TV" with the most probable values falling somewhere around 0.8.

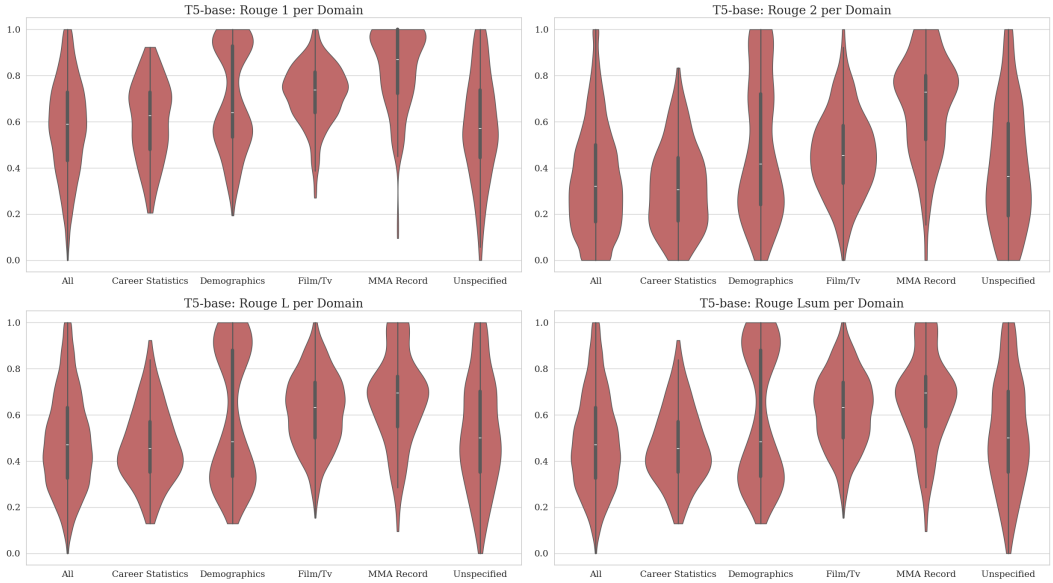


Fig. 7.16: Violin Plot: ROUGE scores of T5-base fine-tuned on ToTTo per domain.

Next, in Tab. 7.13, a sample of machine-generated summaries along with their corresponding annotations is given. The minimum generated sequence length is set to 5 whereas the maximum to 30. The inference time for the whole test set is 20 minutes and for each individual category ranges from 1 to 5 minutes.

Tab. 7.13: T5-base: Generated Summary VS Annotation - ToTTo dataset

Table	Generated Summary	Annotation
0	Daniel Henry Chamberlain was the 76th Governor of South Carolina from December 1, 1874.	Daniel Henry Chamberlin was the 76th Governor of South Carolina from 1874.
1	In 2016, Alma Jodorowsky played the role of Evelyn in the film Kids in Love.	Alma Jodorowsky had the role of Evelyn in the 2016 film Kids in Love.
2	A. J. Hawk had 119 tackles.	In his rookie season, Hawk led with 119 total tackles.

Observations on Tab. 7.13:

- Table 0: It appears that all required information is included in the guided summary. The model also adds the day and month that Chamberlain was elected. According to the corresponding table snippet obtained from the Wikipedia page that lists all governors of South Carolina (Fig. 7.6), the provided information is valid. The sentence does not include any grammatical or syntactical mistakes and gets the point across.
- Table 1: All required information is included in the generated text. The sentence is coherent and does not include any mistakes.
- Table 2: The number of tackles is included in the summary but there is not any information provided about the time-frame. According to the generated summary and the corresponding Wikipedia page, this took place when he was a rookie.

7.1.4 Evaluation and Analysis of Model-Generated Summaries

Below are provided a table (Tab. 7.14) that contains some descriptive statistics regarding the number of words per text along with a boxplot (Fig. 7.17). It appears that Bart-base generates longer summaries in comparison with the other two models and there are outliers present in all groups of texts. The cause of longer summaries produced by Bart-base is probably the fact that chunking is used when encountering input sequences longer than 1,024 tokens.

Tab. 7.14: Number of Words per Text - ToTTo

Model	Average	Median	Standard Deviation	Minimum	Maximum
T5-small	13	12	3	4	27
T5-base	13	13	3	4	26
Bart-base	27	16	53	3	2029
Annotation	15	14	6	3	52

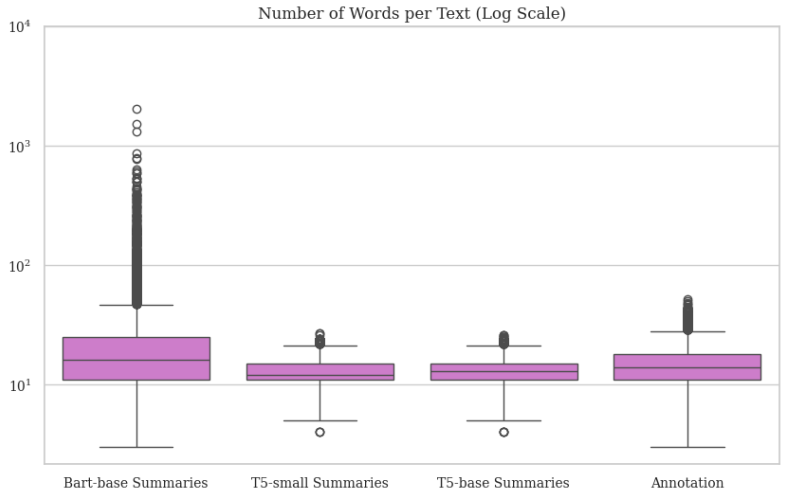


Fig. 7.17: Box Plot: Number of Words per Text - ToTTo

In order to assess text readability the Flesch Reading Ease Score [Dub04] is employed. Flesch Reading Ease formula was developed in the 1940s by Rudolf Flesch and ranges from 0 to 100. The higher the score, the easier a piece of text is to read.

As Table 7.15 shows that both the median and mean range from 74 to 77 which means that the texts are are fairly easy to comprehend. Scores that fall between 70 to 79, according to Theo Acquah [Acq23], signify that 16 to 17-year-old students are able to understand the text.

Tab. 7.15: Flesch Reading Ease Score - ToTTo

Model	Average	Median	Standard Deviation
T5-small	76	77	17
T5-base	76	77	17
Bart-base	75	76	17
Annotation	74	75	18

Fig. 7.18 illustrates the 20 most frequently occurring words in the generated summaries and annotations. The x axis represents the number of words. The bars that are colored in deep purple indicate common occurrences. It is evident that the term "played" is the most frequently used in the summaries of all models, while the predominant word in the annotations is "2010".

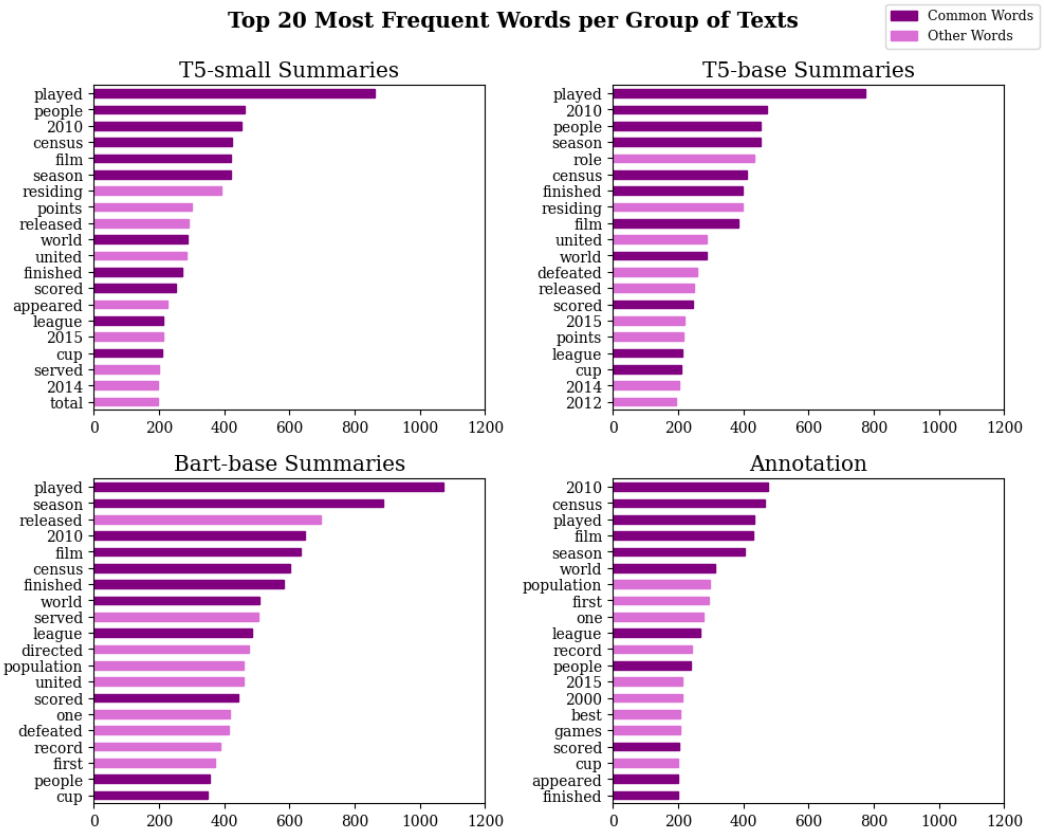


Fig. 7.18: Bar Plot: Top 20 Most Frequent Words per group of Texts - ToTTo

7.2 QTSumm

This section includes information about the training process using the QTSumm dataset and presents the results.

7.2.1 T5-small

The first step is to tokenize the data. All templates and their corresponding annotations are tokenized with a maximum length of 2,048 tokens and 256 tokens accordingly. If the input exceeds the specified maximum length, in both cases, it gets truncated.

Continuing with the training process, after performing hyperparameter tuning and taking into account the available resources, a learning rate of 2e-4 and a batch size of 4 is chosen. The model is trained for 9 epochs, setting the weight decay to 10^{-2} and employing early stopping with patience 3 that monitors validation loss to prevent overfitting. Fig. 7.19 presents the loss curves resulting from training. The duration of the training process is approximately 35 minutes on V100 GPU.

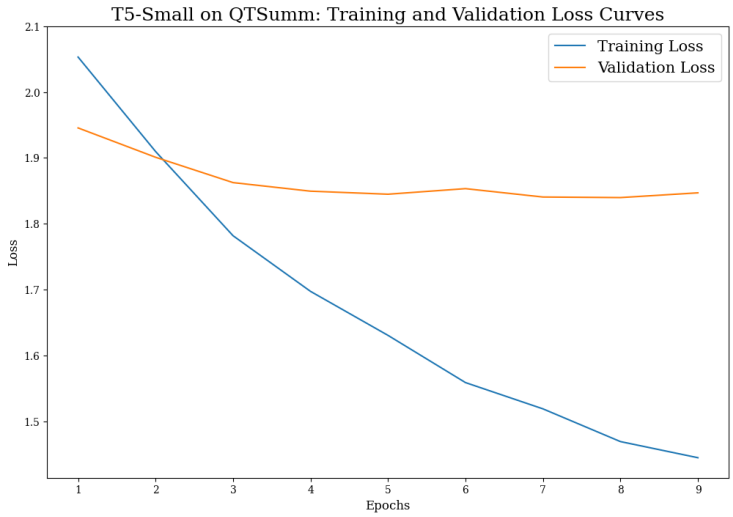


Fig. 7.19: Loss Curves of T5-small trained on the QTSumm dataset.

Next, the model is evaluated on the test set. The results are presented in Tab. 7.16. By taking all metrics under consideration, it seems that the model performs moderately. The Rouge 1 score suggests that the generated summaries share about 52% of their unigrams with the annotation. The Rouge 2 score reveals that the generated summaries share about 27% to 29% of their bigrams with the reference summaries while Rouge-L indicates that the generated summaries have a 38% to 40% overlap in terms of the longest common subsequence of words with the reference summaries. The SacreBleu and Meteor scores indicate that the model generates summaries that are somewhat similar to the annotation in terms of n-gram matches, word order, and word-to-word matches. The standard deviation seems reasonable considering the minimum and maximum values of the chosen metrics.

Tab. 7.16: T5-small Results on QTSumm

Results

Metric	Average	Median	Standard Deviation	Minimum	Maximum
Rouge1	0.5214	0.5245	0.1341	0	0.9655
Rouge2	0.2856	0.2716	0.1360	0	0.8889
Rouge-L	0.3984	0.3846	0.1374	0	0.9655
Rouge-Lsum	0.3983	0.3846	0.1374	0	0.9655
SacreBleu	16.1109	13.5828	11.3419	0	69.997
Meteor	0.398	0.3764	0.1479	0	0.9364

Tab. 7.17: T5-small: Generated Summary VS Annotation - QTSumm dataset

Table	Query	Generated Summary	Annotation
0	Who are the Catholic justices who were appointed by a President Bush and what states are they from?	John Roberts from Maryland, Samuel Alito from New Jersey, and Sonia Sotomayor from New York are the Catholic justices who were appointed by a President Bush. Roberts was appointed in 2005, and Alito was appointed as incumbent in 2005.	John Roberts from Maryland and Samuel Alito from New Zersey were appointed by a President Bush.
1	Summarize the result of the game between Army Cadets and Stevens Tech.	The Army Cadets faced off against Stevens Tech in the first game of the season . The result of the game was a win for the team, with 27 points and a record of 1 - 0.	The Army Cadets played Stevens Tech in the first game of their season on October 5th and won 27-0. This was a shutout win for the Cadets, as they were able to keep their opponents' score at zero while scoring 27 of their own points. Overall, it was a successful start to the 1912 season and they were able to gain momentum going into the next game against Rutgers.
2	What are the differences and similarities in specifications between the 4200 and 4250 models of the HP LaserJet 4000 series?	the HP LaserJet 4000 series has two models, the 4200 and 4250. The 4200 has a CPU speed of 300 MHz, a print resolution of 1200, and a maximum memory of 416 MB. The same as the 4250 model, it is discontinued in June/November 20051, with a processor speed of 460 MHz and an output speed of 43 MHz. The standard memory is 48 MB, and the printer speed is 512 MB .	The HP LaserJet 4200 and 4250 models, both in HP LaserJet 4000 series, have some same and different specifications. In same parts, both models come out in November 2002 and stop in June/November 20051. They have print resolution of 1200 DPI. Also, both printers can make the memory bigger, up to 512 MB. But, there are some different things between these two models. The LaserJet 4200 has 300 MHz CPU speed, but 4250 has faster 460 MHz CPU. Also, 4200 model can print 35 pages in one minute (PPM), but 4250 can print a little faster, 43 PPM.

Tab. 7.17 includes examples of model generated summaries and their corresponding human annotations along with the query they need to answer. The minimum output length is set to 1 and the maximum is set to 150. Regarding inference time, 15 minutes are needed to generate all summaries for the tables of the test set.

Observations on Tab. 7.17:

- For Table 0, the generated summary provides a relevant answer to the query. The model accurately incorporated the names of John Roberts and Samuel Alito, along with their respective states, in the generated summary as shown in Fig. 7.20. The name of Sonia Sotomayor from New York is also included but this Catholic justice was appointed by President Obama and not President Bush. Additionally, an extra piece of information has been added to the generated summary, specifying the year of appointment. While the information regarding John Roberts is accurate, it should be noted that the information for Samuel Alito is not. In general the generated text surpasses the length of the reference text providing more information than requested. The generated summary is coherent and there no grammatical or syntactical mistakes encountered. Finally, a typographic oversight is present in the annotation where "New Jersey" is mistakenly written as "New Zersey".

Name	State	Birth	Death	Year appointed	Left office	Appointed by	Reason for termination
Roger B. Taney	Maryland	1777	1864	1836	1864	Jackson	death
Edward Douglass White	Louisiana	1845	1921	1894	1921	Cleveland (associate) Taft (chief)	death
Joseph McKenna	California	1843	1926	1898	1925	McKinley	retirement
Pierce Butler	Minnesota	1866	1939	1923	1939	Harding	death
Frank Murphy	Michigan	1890	1949	1940	1949	F. Roosevelt	death
Sherman Minton	Indiana	1890	1965	1949	1956	Truman	death
William J. Brennan, Jr.	New Jersey	1906	1997	1956	1990	Eisenhower	death
Antonin Scalia	New Jersey	1936	2016	1986	2016	Reagan	death
Anthony Kennedy	California	1936	living	1988	2018	Reagan	retirement
Clarence Thomas	Georgia	1948	living	1991	incumbent	G. H. W. Bush	—
John Roberts	Maryland	1955	living	2005	incumbent	G. W. Bush	—
Samuel Alito	New Jersey	1950	living	2006	incumbent	G. W. Bush	—
Sonia Sotomayor	New York	1954	living	2009	incumbent	Obama	—
Brett Kavanaugh	Washington, D.C.	1965	living	2018	incumbent	Trump	—

Fig. 7.20: Table 0 of Test Set - QTSumm Dataset

- Regarding Table 1, the instruction is to summarize the game between Army Cadets and Stevens Tech. The model provides a relevant summary stating that the it was the first game of the season and that the Cadets won with 27 points. Then, it mentions a record of 1-0 that according to 7.21 is valid. The summary does not mention anything about the score of the opponent or the date that the game took place. Finally, note that in this example the generated text is shorter in length than the annotation but it is coherent and does not include any grammatical or syntactical mistakes.

Game	Date	Opponent	Result	Black Knights Points	Opponents	Record
1	Oct 5	Stevens Tech	Win	27	0	1 - 0
2	Oct 12	Rutgers	Win	19	0	2 - 0
3	Oct 19	Yale	Loss	0	6	2 - 1
4	Oct 26	Colgate	Win	18	7	3 - 1
5	Nov 9	Carlisle Indian	Loss	6	27	3 - 2
6	Nov 16	Tufts University	Win	15	6	4 - 2
7	Nov 23	Syracuse	Win	23	7	5 - 2

Fig. 7.21: Table 1 of Test Set - QTSumm Dataset

- The query given for Table 2 (Fig. 7.22) requests information about the differences and similarities in specifications between the 4200 and 4250 models of HP LaserJet 4000 series. With a first glance the generated summary and the reference summary seem to have approximately the same length. This is a more challenging task as it requires greater reasoning capabilities than the previous two. The generated summary states an "output speed of 43 MHz" for the 4250 model. Here the given number is valid but the unit used is not as the printing speed is measured in pages per minute (PPM). In general there are many mistakes and the text seems to be quite confusing.

Model	Introduction	Discontinued	CPU Speed	Print resolution (DPI)	Print speed (PPM)	Standard memory	Maximum memory
4000	November 1997	May 1999	100 MHz	1200	17	4 MB	100 MB
4050	May 1999	November 2001	133 MHz	1200	17	8 MB	192 MB
4100	March 2001	February 2003	250 MHz	1200	25	16 MB	256 MB
4200	November 2002	June/November 20051	300 MHz	1200	35	48 MB	416 MB
4240	? - Please add details on this model	? - Discontinued	460 MHz	1200	40	64 MB	512 MB
4250	November 2002	June/November 20051	460 MHz	1200	43	48 MB	512 MB
4300	December 2002	June 2005	350 MHz	1200	45	64 MB	416 MB
4350	November 2002	June 2005	460 MHz	1200	52	80 MB	512 MB

Fig. 7.22: Table 2 of Test Set - QTSumm Dataset

7.2.2 Bart-base

Continuing with Bart-base, the first step again is tokenization. All templates and their corresponding annotations are tokenized with a maximum length of 1,024 tokens and 256 tokens accordingly. If the input exceeds the specified maximum length, in both cases, it gets truncated.

Proceeding with the training process, after performing hyperparameter tuning and considering the available resources, a learning rate of 2e-4 and a batch size of 4 is chosen. The model is trained for 6 epochs, setting weight decay to 10^{-2} and employing early stopping with patience 3 that monitors validation loss to aid avoid overfitting. The training process lasted around 1 hour on V100 GPU. Fig. 7.23 illustrates the loss curves resulting from training.

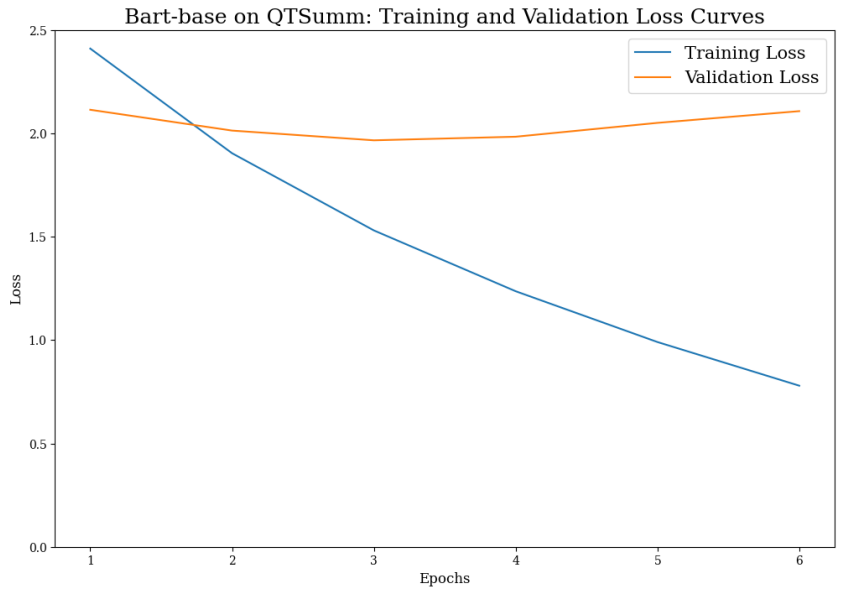


Fig. 7.23: Loss Curves of Bart-base trained on the QTSumm dataset.

Following that, the model is evaluated on the test set. The results are presented in Tab. 7.18. Upon examining all metrics, it appears that the model's performance is moderate. The Rouge-1 score implies that approximately 51% to 52% of the unigrams in the generated summaries align with those in the reference texts. The Rouge 2 score reveals that the generated summaries share about 26% to 28% of their bigrams with the reference summaries while Rouge-L indicates that the generated summaries have a 37% to 40% overlap in terms of the longest common subsequence of words with the reference summaries. The SacreBleu and Meteor scores indicate that the model generates summaries that are somewhat similar to the annotation in terms of n-gram matches, word order, and word-to-word matches. The standard deviation seems reasonable considering the minimum and maximum values that the chosen metrics can take.

Tab. 7.18: Bart-base Results on QTSumm

Results

Metric	Average	Median	Standard Deviation	Minimum	Maximum
Rouge1	0.5153	0.5127	0.1376	0	0.9796
Rouge2	0.2783	0.2621	0.1387	0	0.9362
Rouge-L	0.3951	0.3739	0.1375	0	0.9796
Rouge-Lsum	0.3951	0.3750	0.1375	0	0.9796
SacreBleu	16.2992	13.9352	11.5794	0	89.854
Meteor	0.4000	0.3801	0.1512	0	0.9954

Table 7.19 presents a sample of model-generated summaries, accompanied by their respective annotations and queries. The generated output lengths range from a minimum of 1 to a maximum of 150. The time required for inference is 15 minutes for the whole test set.

Tab. 7.19: Bart-base: Generated Summary VS Annotation - QTSumm dataset

Table	Query	Generated Summary	Annotation
0	Who are the Catholic justices who were appointed by a President Bush and what states are they from?	The Catholic Supreme Court appointed two Catholic justices who were appointed by a President Bush: John Roberts from Maryland and Samuel Alito from New Jersey. John Roberts was born in 1955 and died in 2005. He was incumbent of the court and did not manage to terminate his term. SamuelAlito, born in 1950, was incumbent for the duration of his tenure.	John Roberts from Maryland and Samuel Alito from New Zersey were appointed by a President Bush.
1	Summarize the result of the game between Army Cadets and Stevens Tech.	The Army Cadets faced off against Stevens Tech on October 5th. The Cadets won the game with a score of 27-0. This put them at a 1-0 record.	The Army Cadets played Stevens Tech in the first game of their season on October 5th and won 27-0. This was a shutout win for the Cadets, as they were able to keep their opponents' score at zero while scoring 27 of their own points. Overall, it was a successful start to the 1912 season and they were able to gain momentum going into the next game against Rutgers.
2	What are the differences and similarities in specifications between the 4200 and 4250 models of the HP LaserJet 4000 series?	The HP LaserJet 4000 series has many different specifications and differences in specifications between the 4200 and 4250 models. The 4200 has a maximum memory of 1200 GB, while the 4250 has a minimum memory of 450 GB. The 250 has more big maximum memory than 4200, with maximum memory being 512 GB.	The HP LaserJet 4200 and 4250 models, both in HP LaserJet 4000 series, have some same and different specifications. In same parts, both models come out in November 2002 and stop in June/November 20051. They have print resolution of 1200 DPI. Also, both printers can make the memory bigger, up to 512 MB. But, there are some different things between these two models. The LaserJet 4200 has 300 MHz CPU speed, but 4250 has faster 460 MHz CPU. Also, 4200 model can print 35 pages in one minute (PPM), but 4250 can print a little faster, 43 PPM.

Observations on Tab. 7.19:

- Regarding Table 0, the generated summary is longer than the reference. The requested information is present in the generated text along with some additional details. John Roberts indeed was born in 1,955 but he is still alive according to Fig. 7.20 and still on duty. Moreover, in the generated summary it is stated Samuel Alito was born in 1,950 which is valid and is still incumbent. However, the use of the past simple tense to convey the information suggests that the model may be assuming their decease.
- Moving on to Table 1, the generated summary, albeit concise, provides the requested information in three coherent sentences. According to Fig. 7.21, all the statements are valid.
- Table 2 is more challenging for the model as it requires higher reasoning capabilities in comparison with the other two. Although the text is not very coherent and includes many inaccuracies (Fig. 7.22), it became clear to the model that the request is a comparison between the two models of 4000 HP LaserJet series.

7.2.3 T5-base

To begin with, all templates and their corresponding annotations undergo tokenization with a maximum length of 2,048 tokens and 256 tokens, respectively. Should the input exceed the designated maximum length in either case, truncation is applied.

Moving on to the training process, hyperparameter tuning is performed to aid select the optimal learning rate and batch size. The empirically selected values are 2e-06 and 4 respectively. The model is trained for 2 epochs, applying a weight decay of 10^{-1}. The training process lasted 1 hour on V100 GPU.

Continuing with the results, according to the ROUGE scores present in Tab. 7.20, the model's performance could be deemed moderate. Regarding SacreBleu and Meteor, included in the aforementioned table, the performance seems to be on the lower end.

Tab. 7.20: T5-base Results on QTSumm

Results

Metric	Average	Median	Standard Deviation	Minimum	Maximum
Rouge1	0.4358	0.4348	0.1189	0	0.8462
Rouge2	0.2144	0.2035	0.1090	0	0.7500
Rouge-L	0.3164	0.3060	0.1070	0	0.8462
Rouge-Lsum	0.3165	0.3061	0.1070	0	0.8462
SacreBleu	10.2954	7.9374	8.4059	0	69.3098
Meteor	0.3281	0.3059	0.1400	0	0.8452

The time needed to generate the summaries for the whole test set is approximately 30 minutes. The minimum output length is set to 1 and the maximum is set to 150. Tab. 7.21 provides 3 examples of generated summaries along with their respective queries and annotations.

Tab. 7.21: T5-base: Generated Summary VS Annotation - QTSumm dataset

Table	Query	Generated Summary	Annotation
0	Who are the Catholic justices who were appointed by a President Bush and what states are they from?	who are the Catholic justices who were appointed by a President Bush and what states are they from?	John Roberts from Maryland and Samuel Alito from New Zersey were appointed by a President Bush.
1	Summarize the result of the game between Army Cadets and Stevens Tech.	Army Cadets play Stevens Tech in a game on Oct 5 . the game was won by the army, with a score of 27 points and a draw of 1 - 0. The result was a win for the army cadets .	The Army Cadets played Stevens Tech in the first game of their season on October 5th and won 27-0. This was a shutout win for the Cadets, as they were able to keep their opponents' score at zero while scoring 27 of their own points. Overall, it was a successful start to the 1912 season and they were able to gain momentum going into the next game against Rutgers.
2	What are the differences and similarities in specifications between the 4200 and 4250 models of the HP LaserJet 4000 series?	HP LaserJet 4000 series has two models: 4200 and 4250. each has a speed of 460 MHz and a print speed of 40 Mbps. both models have similar specifications, with a maximum of 512 MB of memory . the 4200 series has the fastest CPU speed at 450 MHz .	The HP LaserJet 4200 and 4250 models, both in HP LaserJet 4000 series, have some same and different specifications. In same parts, both models come out in November 2002 and stop in June/November 20051. They have print resolution of 1200 DPI. Also, both printers can make the memory bigger, up to 512 MB. But, there are some different things between these two models. The LaserJet 4200 has 300 MHz CPU speed, but 4250 has faster 460 MHz CPU. Also, 4200 model can print 35 pages in one minute (PPM), but 4250 can print a little faster, 43 PPM.

Observations on Tab. 7.21:

- Regarding Table 0, the model did not provide the sought-after information, instead it returned the query.
- The generated summary of Table 1 includes the necessary information according to Fig. 7.21; however, there is no consistency in the tense used as it uses Present Simple in the beginning but Past Simple latter on. Additionally, the first letter of the first word in the second sentence should be uppercase, and the letter "A" in 'Army' should also be capitalized since it refers to the 'Army Cadets' team.
- In the summary of Table 2, the consistent problem of neglecting to capitalize the first letter of each sentence persists. Although the model appears to recognize this

as a comparison between two printers, numerous errors are present in the generated text as one can verify by looking at Fig. 7.22. One such error involves the use of Mbps to measure printing speed.

7.2.4 Evaluation and Analysis of Model-Generated Summaries

Tab. 7.22 contains some descriptive statistics regarding the number of words per text. The generated summaries appear to be shorter in length than the reference texts as the mean and median indicate. The highest standard deviation belongs to the Annotation.

Tab. 7.22: Number of Words per Text - QTSumm

Model	Average	Median	Standard Deviation	Minimum	Maximum
T5-small	45	41	19	12	109
T5-base	45	44	13	13	96
Bart-base	48	44	21	12	117
Annotation	58	53	28	1	138

Next, to get an idea about how the number of words per text is distributed, Fig. 7.24 is provided. It appears that all categories have at least one outlier. For T5-small, approximately 50% of the generated summaries fall within the range of 30 to 60 words. The number of words that 50% of T5-base generated summaries consist of fall between 35 and 50. As far as Bart-base is concerned, the 50% of the generated summaries is comported by 30 to 60 words. Lastly, the Annotations encompass 40 to 80 words for 50% of the cases.

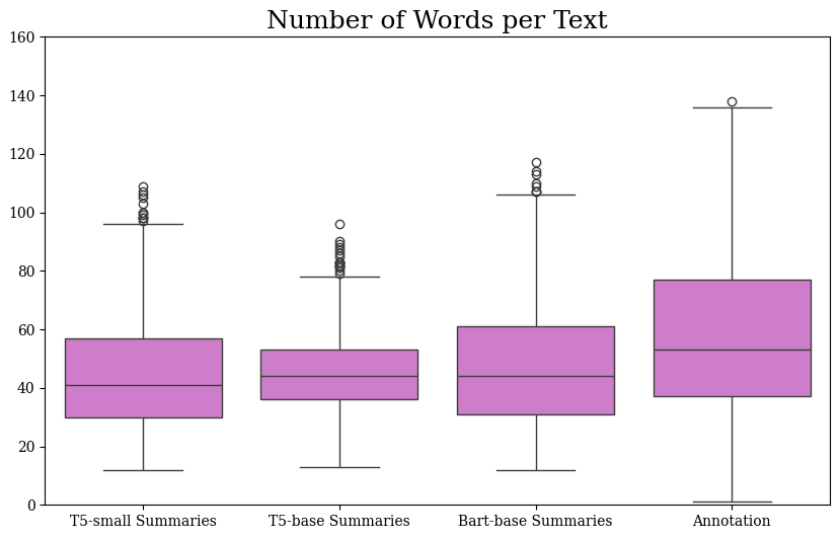


Fig. 7.24: Box Plot: Number of Words per Text - QTSumm

Tab. 7.23 presents the results of Flesch Reading Ease Score [Dub04]. As one can observe the Annotations have a lower score compared to the model generated summaries. This indicates that the reference summaries are harder to understand compared to the generated summaries. A score that ranges from 50 to 59 signifies that the text is fairly difficult to read, as Theo Acquah states [Acq23]. On the other hand, the summaries that the models generated scored between 70 and 79 making them fairly easy to read.

Tab. 7.23: Flesch Reading Ease Score - QTSumm

Model	Average	Median	Standard Deviation
T5-small	72	74	15
T5-base	74	76	14
Bart-base	72	74	15
Annotation	58	53	28

Fig. 7.25 illustrates the 20 most frequently occurring words in the generated summaries and annotations. The x axis represents the number of words. The bars that are colored in deep purple indicate common occurrences.

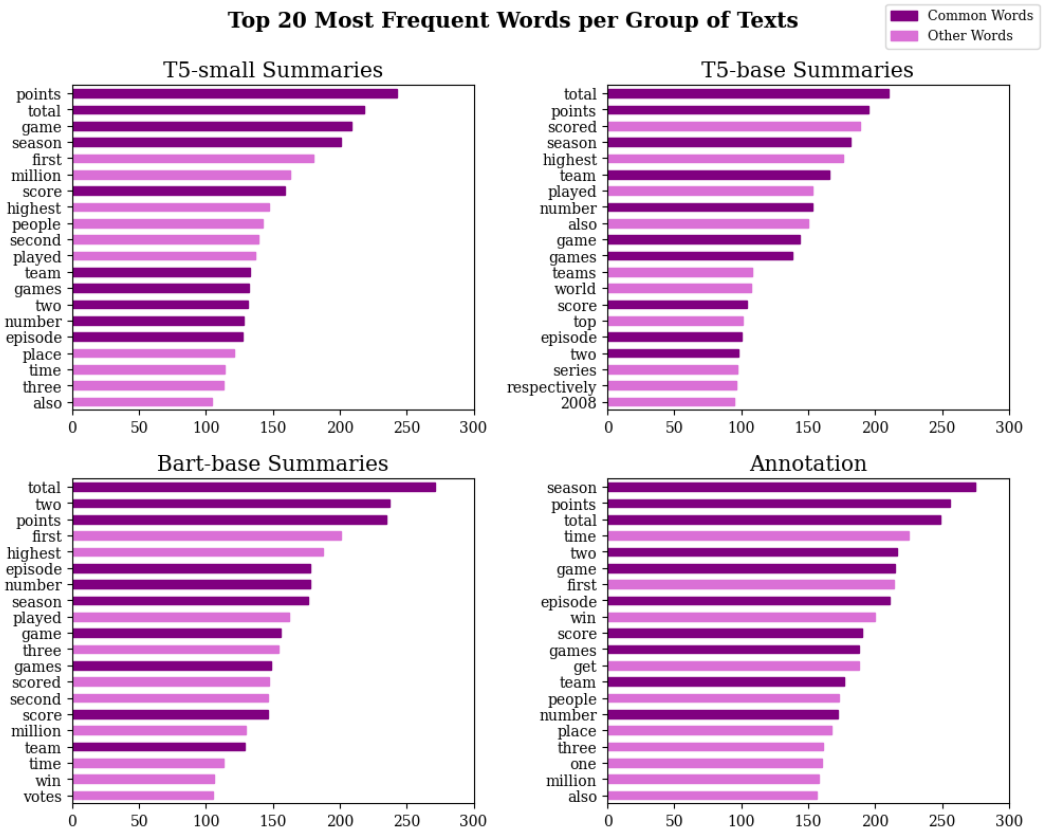


Fig. 7.25: Bar Plot: Top 20 Most Frequent Words per group of Texts - QTSumm

Comparative Data Analysis

8

In this section, a comparative data analysis is conducted to evaluate the performance of the models against each other and the corresponding benchmark.

8.1 ToTTo

The highest BLEU score belongs to Bart-base followed by T5-base as Fig. 8.1 shows. Even though T5-base was trained for only one epoch, it outperformed T5-small and performed almost as good as Bart-base. Note that the latter two were trained for 4 epochs. All models outperformed NARRATABLE.

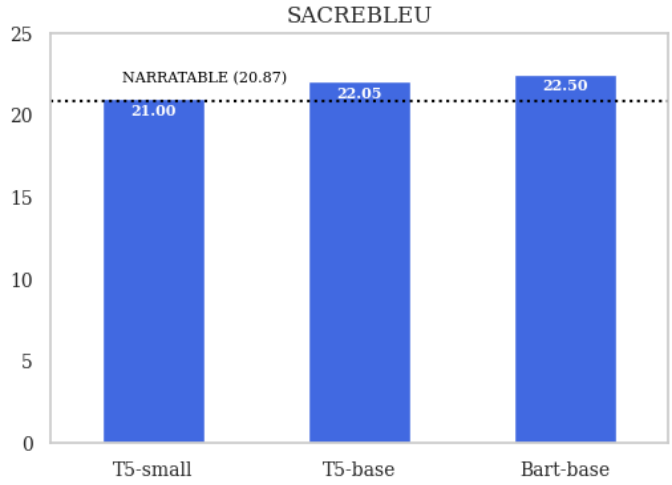


Fig. 8.1: Bar Plot: Average BLEU scores of all models - ToTTo.

According to Fig. 8.2, the highest METEOR score again belongs to Bart-base followed by T5-base. Both models outperformed NARRATABLE while T5-small performed at the same level. The higher METEOR scores that BART-base exhibits compared to the T5 variations employed, suggests a capacity to generate summaries that align more closely with the semantic content and word order of the reference text. The model seems to be able to capture key ideas and phrases from input table templates, presenting them in a generally coherent and well-organized manner, while maintaining the essential order of information in the summaries. This claim can also be supported by examining the Generated Summary - Reference Summary pairs provided in Tab. 7.9 and by taking into account the Flesch Reading Ease Score of the model's summaries that categorizes them as "fairly easy" to comprehend.

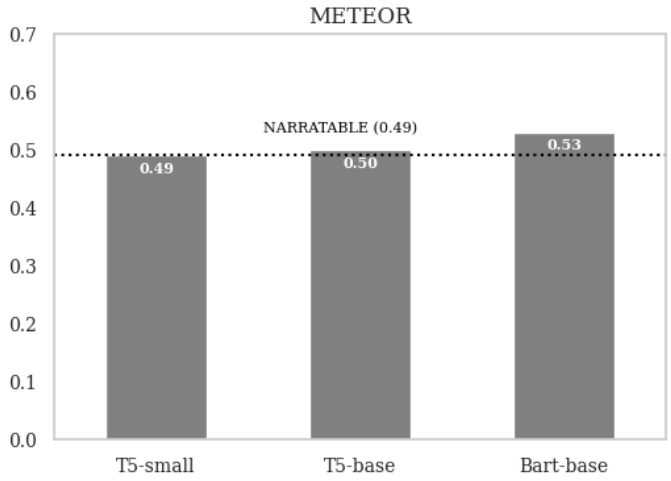


Fig. 8.2: Bar Plot: Average METEOR scores of all models - ToTTo.

Regarding the ROUGE scores, for ROUGE 1 the highest performance is observed for T5-base followed by T5-small as indicated by Fig. 8.3. Both models outperformed NARRATABLE slightly whereas Bart-base performed on the same level. A higher ROUGE 1 score signifies a greater number of overlapping unigrams between the annotations and the corresponding generated summaries.

Regarding ROUGE 2, all models seem to outperform NARRATABLE slightly with the higher score belonging to T5-base and Bart-base. A higher ROUGE 2 score signifies a greater number of overlapping bigrams between the annotations and the corresponding generated summaries.

Proceeding to ROUGE L, it appears that only T5-base slightly outperformed NARRATABLE. T5-small performed on the same level while Bart-base scored lower.

Finally, the highest score of ROUGE Lsum holds T5-base, sightly outperforming NARRATABLE. T5-small and Bart-base perform on the same level.

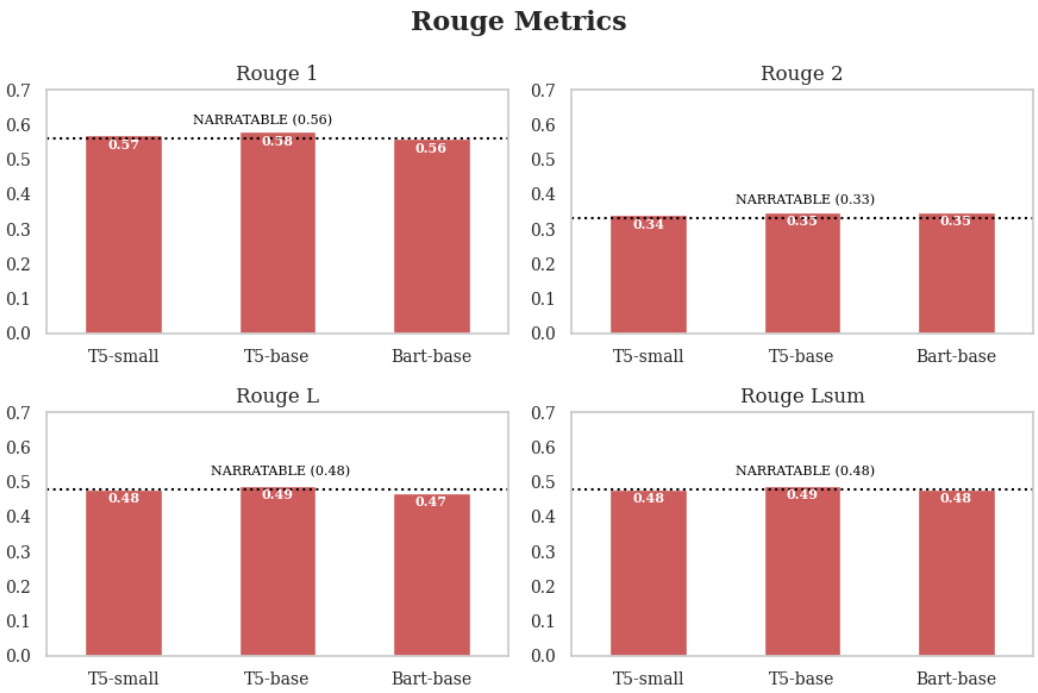


Fig. 8.3: Bar Plot: Average ROUGE scores of all models - ToTTo.

The model with the highest scores across all metrics is Bart-base; however, the same model also exhibits the highest standard deviation, as shown in Table 8.1. This indicates a higher degree of performance variability, resulting in summaries that may deviate from the expected quality, either improving or worsening. However, the difference in standard deviation is not significant.

Tab. 8.1: All Domains: Standard Deviation - ToTTo

Standard Deviation

Metric	T5-small	T5-base	Bart-base
Rouge1	0.20	0.21	**0.23**
Rouge2	0.23	0.23	**0.24**
Rouge-L	0.21	0.21	**0.23**
Rouge-Lsum	0.21	0.21	**0.23**
SacreBleu	19.56	20.19	**22.03**
Meteor	0.22	**0.23**	0.22

8.2 QTSumm

It seems that in terms of SacreBleu the models do not reach the bench mark since T5-large [Zha+23] achieves higher scores. This can be justified by the fact that the researchers used the larger versions of all models that they experimented with and trained them for more epochs. The lowest performance is observed for T5-base but it is important to keep in mind that the model is trained for only two epochs. T5-small and Bart-base perform at the same level with Bart-base slightly surpassing the aforementioned T5 variation. Fig. 8.4 provides a visual representation.

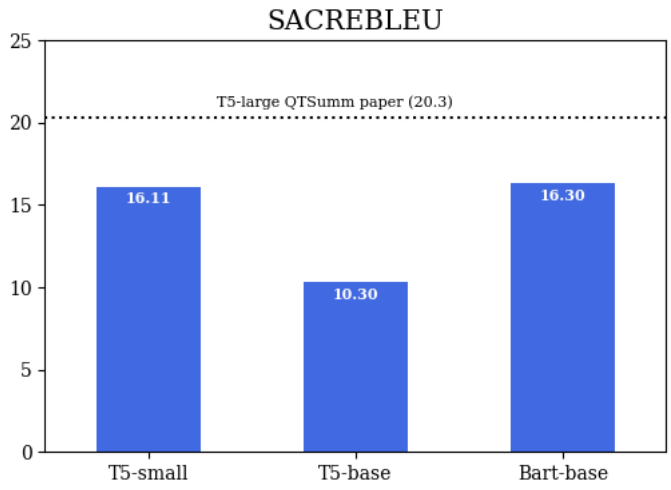


Fig. 8.4: Bar Plot: SacreBleu scores of all models employed along with the selected benchmark - QTSumm.

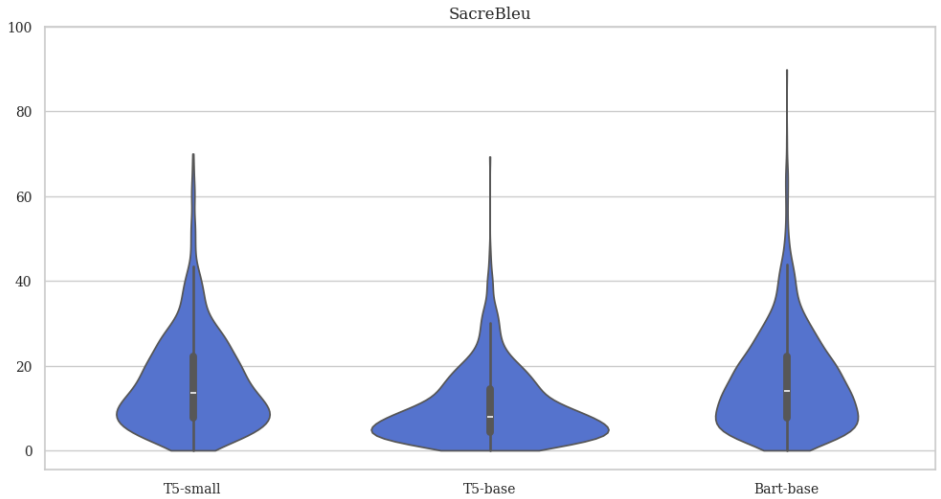


Fig. 8.5: Violin Plot: SacreBleu scores of all models - QTSumm.

Fig. 8.5 includes the violin plots of the SacreBleu scores for each model. For T5-small it appears that the most probable values are around 10 with the lower value being 0 and the highest being around 70. For T5-base the most probable values are gathered around 5 with the lower score being 0 and the highest score achieved being around 70. Finally, the most

probable values for Bart-base appear around 10 with 0 being the lowest score observed and 90 the highest.

Meteor scores seem to be more promissing as T5-small and Bart-base reached the performance of T5-large [Zha+23] while for T5-base a lower score is observed as seen in 8.6.

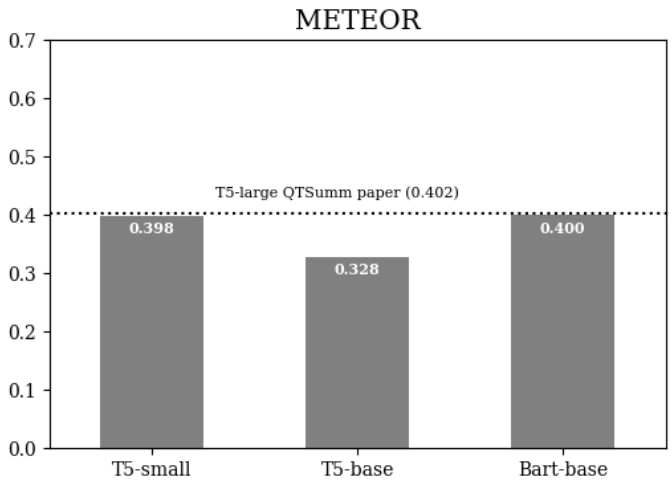


Fig. 8.6: Bar Plot: Meteor scores of all models employed along with the selected benchmark - QTSumm.

The violin plots for Meteor scores (Fig. 8.7) exhibit a similar shape, with Bart-base achieving scores near the upper limit in some instances. For T5-small, the scores range from 0 to approximately 0.95, with the most probable values centered around 0.3. T5-base scores also range from 0 to 0.8, with the most probable values clustered around 0.3. In contrast, Bart-base achieves higher scores, ranging from 0 to 1, with the most probable values observed around 0.4.

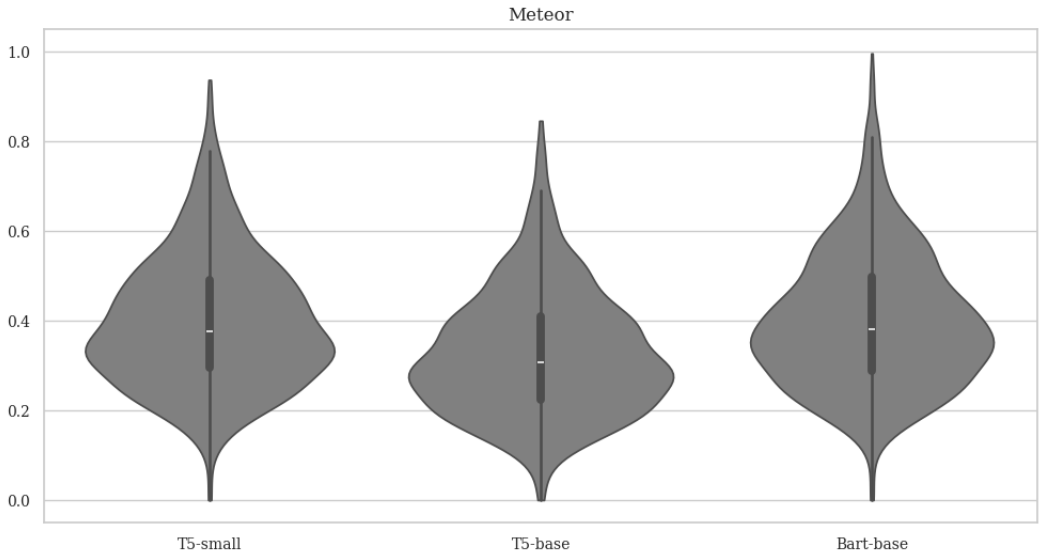


Fig. 8.7: Violin Plot: Meteor scores of all models - QTSumm.

Moving on to ROUGE-L, the performance of T5-small and Bart-base seems to slightly surpass the one of T5-large [Zha+23] while T5-base demonstrates comparatively lower performance.

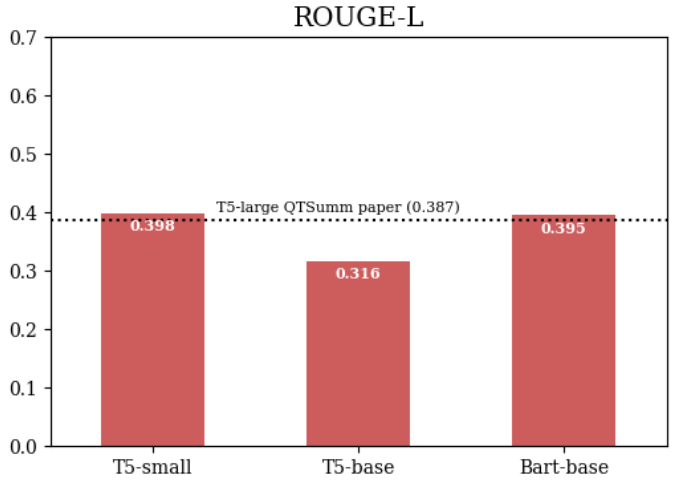


Fig. 8.8: Bar Plot: ROUGE-L scores of all models employed along with the selected benchmark - QTSumm.

Fig. 8.9 includes the violin plots of the ROUGE metrics. Regarding ROUGE 1 the most probable values appear betweeen 0.4 and 0.6, for ROUGE 2 between 0.1 and 0.4 and finally for ROUGE L and ROUGE Lsum between 0.2 and 0.5

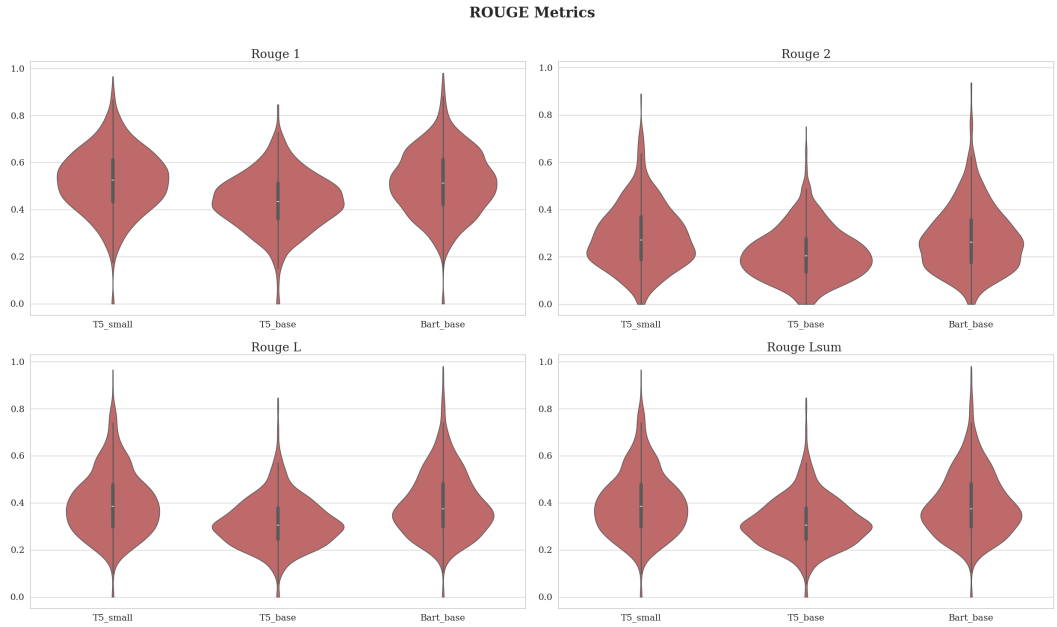


Fig. 8.9: Violin Plot: ROUGE scores of all models - QTSumm.

Tab. 8.2 gathers the information relevant the standard deviation present in all metrics of all models trained on the QTSumm dataset. The highest values appear in Bart-base followed by T5-small. Higher standard deviation values imply that the performance of the model varies.

Tab. 8.2: Standard Deviation - QTSumm

Standard Deviation

Metric	T5-small	T5-base	Bart-base
Rouge1	0.1341	0.1189	**0.1376**
Rouge2	0.1360	0.1090	**0.1387**
Rouge-L	0.1374	0.1070	**0.1375**
Rouge-Lsum	0.1374	0.1070	**0.1375**
SacreBleu	11.3419	8.4059	**11.5794**
Meteor	0.1479	0.1400	**0.1512**

In this task, it appears that the T5-base model is affected by the limited number of epochs it has been trained on, resulting in lower performance compared to its counterparts that have undergone more extensive training.

Conclusions

9

The model that achieves the highest scores on the entire test-set in both datasets is Bart-base with the difference being more noticeable on ToTTo. However, this model exhibits the highest standard deviation across almost all metrics resulting in summaries that vary in terms of quality. If consistency is more important, for the QTSumm dataset, T5-small could be a better option as the performance is similar making the effects of higher standard deviation more significant. Regarding the ToTTo dataset, the difference in performance of Bart-base and T5-base is marginal as depicted in Fig. 8.1 and Fig. 8.2. Therefore, the difference in standard deviation is considered negligible.

As the results for the ToTTo dataset attest,the strength of the selected T5 variations lies in generating high quality summaries for tables that come from the "MMA Record" domain whereas Bart-base is better at generating summaries for tables that come from the "Demographics" category. However, the domain of "Demographics" seems to have the highest standard deviation in all cases implying a greater degree of performance inconsistency compared to other domains.

Both in ToTTo and QTSumm it is observed that the SacreBleu scores are lower than the METEOR scores. SacreBleu which is based on BLEU is a precision-oriented metric whereas METEOR considers both precision and recall. A lower BLEU score suggests that there is less agreement between the generated text and the reference in terms of word sequences and n-grams. This could be due to the generation of incorrect words, phrases, or word order, leading to lower precision. Especially for QTSumm, this appears to be true if the Generated Text- Annotation pairs of Tab. 7.17, Tab.7.19 and Tab. 7.21 are taken under consideration as there are observed many inaccuracies and the generated summaries are often longer or shorter than the reference text. Moreover, as BLEU examines exact n-gram matches, if a summary has the same meaning with the reference text but does not use the exact same words, the score will be lower.

METEOR takes into account synonyms and word variations, as well as stemming. If the generated text uses synonyms or stems words appropriately, it can contribute to a higher METEOR score. Moreover, this metric considers the word order in the generated text compared to the reference. If the generated text maintains a coherent and correct word order, it can receive a higher METEOR score. As this metric has higher correlation with human judgement than BLEU [BL05], it provides a more accurate representation of the models' performance.

Additionally, in some cases, the model-generated summaries accurately incorporate the highlighted information while the reference summaries may not encompass all the details. For example, in Tab. 7.13, the model added the exact date that Daniel Henry Chamberlain

was elected but the reference summary includes only the year even though the highlighted cell contains the complete date. This leads to a lower SacreBleu and METEOR, as it negatively affects the precision, even though all the provided information is valid (according to Fig. 7.6) and conveyed in a coherent sentence. The effects are more prominent in SacreBleu as it is a precision-based metric.

Overall, the results attest that the three models perform better on the ToTTo dataset compared to QTSumm. This is not surprising, given that the task is inherently more demanding. ToTTo is a single-sentence generation task where the focus is on generating fluent and faithful descriptions using provided table regions to encourage guided text generation. On the other hand, QTSumm is a query-focused summarization task where the goal is to generate paragraph-long summaries that answer the user's questions over a table of interest. QTSumm requires higher reasoning abilities since the queries range from simple table summarization to comparison between values etc. It is also important to remember that the input templates of QTSumm are given to the models without previously performing any text cleaning.

In both cases, the input data involve the conversion of the table of interest into text. Regarding ToTTo templates, the information from the highlighted cells is integrated, whereas for QTSumm templates, the data from the table rows containing the requested information is included. Despite adding the highlighted rows to the QTSumm templates, it falls short of replicating the presentation of highlighted cells in ToTTo templates. This difference arises from the fact that the latter specifically provides only the essential data. The templates of ToTTo also include relevant metadata providing insights about the table's subject, the title of the page of origin, etc. Those additional data probably also lead to better results. Finally, in ToTTo the instruction is "Summarize:" in all cases while in QTSumm is the given queries.

10 Limitations and Future Work

Besides Bart-base, T5-small and T5-base, Llama2 7B Chat [Tou+23] was fine-tuned using Low-rank Adaptation (LORA) [Hu+21] on the ToTTo dataset. LORA does not introduce additional layers or require prompt fine-tuning, but instead modifies the values of parameters. The additional weight changes are stored in a matrix whereas the all the pre-trained model weights remain the same. The model was trained for 3 epochs and the training process lasted around 7 hours using a V100 GPU.

Even though the generated summaries seemed to be very promising the necessary resources for inference surpass 40 GB of GPU RAM. To tackle this issue, an attempt was made to split the test set into batches the size of which ranged from 2 to 20 and the model was loaded in 4 bit mode to save memory and increase the processing speed. Since the test set comprises of 7,700 tables, it would be impossible to generate all summaries in a reasonable time span. For this reason, there are not any results available. The next step in this case is to obtain enough GPU RAM to generate all test set summaries and publish the results.

Tab. 10.1 showcases examples of generated summaries resulting from Llama2 7B Chat and their corresponding annotations.

Tab. 10.1: Llama2 7B Chat: Generated Summary VS Annotation - ToTTo dataset

Table	Generated Summary	Annotation
0	Daniel Henry Chamberlain was the 76th Governor of South Carolina.	Daniel Henry Chamberlin was the 76th Governor of South Carolina from 1874.
1	Sure! Here is the summary of the text in one sentence: In 2016, Alma Jodorowsky played Evelyn in the film Kids in Love.	Alma Jodorowsky had the role of Evelyn in the 2016 film Kids in Love.
2	In 2006, A.J. Hawk played in 16 games, combined tackles were 119, assisted tackles were 37, sacks were 3.5, forced fumbles were 1, fumble recoveries were 2, interceptions were 2, passes defensed were 8, and longest interception return was 25.	In his rookie season, Hawk led with 119 total tackles.
3	Peter II the Simple (Pêr II) ruled from 1450 to 1457, succeeded by Arthur III the Justicier (Arzhur III) who ruled from 1457 to 1458.	At the very end of his life, Arthur III became duke of Brittany, succeeding Peter II.

Bibliography

[Ach+23] OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, et al. "GPT-4 Technical Report". In: 2023.

[Acq23] Theo Acquah. *A Guide to Mastering Readability: The Flesch Reading Ease Score*. Nov. 2023. URL: https://medium.com/@theobee.acquah/a-guide-to-mastering-readability-the-flesch-reading-ease-score-2c67ba7f2a09.

[And+22] Ewa Andrejczuk, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, and Yasemin Altun. *Table-To-Text generation and pre-training with TabT5*. 2022. arXiv: 2210.09162 [cs.CL].

[BL05] Satanjeev Banerjee and Alon Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments". In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ed. by Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72.

[Cha+21] Clayton Chapman, Lars Hillebrand, Marc Stenzel, et al. *Towards Generating Financial Reports From Table Data Using Transformers*. Dec. 2021.

[Che+20] 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*. Ed. by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault. Online: Association for Computational Linguistics, July 2020, pp. 7929–7942.

[Che+21a] Xinyun Chen, Petros Maniatis, Rishabh Singh, et al. *SpreadsheetCoder: Formula Prediction from Semi-structured Context*. 2021. arXiv: 2106.15339 [cs.SE].

[Che+21b] Zhiyu Chen, Wenhu Chen, Charese Smiley, et al. *FinQA: A Dataset of Numerical Reasoning over Financial Data*. 2021. arXiv: 2109.00122 [cs.CL].

[Che+22] Miao Chen, Xinjiang Lu, Tong Xu, et al. *Towards Table-to-Text Generation with Pretrained Language Model: A Table Structure Understanding and Text Deliberating Approach*. 2022. arXiv: 2301.02071 [cs.CL].

[Chu+22] Hyung Won Chung, Le Hou, Shayne Longpre, et al. *Scaling Instruction-Finetuned Language Models*. 2022. arXiv: 2210.11416 [cs.LG].

[Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. arXiv: 1810.04805 [cs.CL].

[Dos21] Ketan Doshi. *Foundations of NLP Explained — Bleu Score and WER Metrics*. May 2021. URL: https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b.

[Dub04] William Dubay. "The Principles of Readability". In: *CA* 92627949 (Jan. 2004), pp. 20–22.

[Gon+19] Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. *Table-to-Text Generation with Effective Hierarchical Encoder on Three Dimensions (Row, Column and Time)*. 2019. arXiv: 1909.02304 [cs.CL].

[Her+20] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. "TaPas: Weakly Supervised Table Parsing via Pre-training". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

[Hu+21] Edward J. Hu, Yelong Shen, Phillip Wallis, et al. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.CL].

[Jia+22] Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. "OmniTab: Pretraining with Natural and Synthetic Data for Few-shot Table-based Question Answering". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, July 2022, pp. 932–942.

[Jia+23] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. *Mistral 7B*. 2023. arXiv: 2310.06825 [cs.CL].

[Kha+22] Vikas Khare, Cheshta Khare, Savita Nema, and Prashant Baredar. "Chapter 5 - Assessment of solar energy system by probability and sampling distribution". In: *Decision Science and Operations Management of Solar Energy Systems*. Ed. by Vikas Khare, Cheshta Khare, Savita Nema, and Prashant Baredar. Academic Press, 2022, pp. 145–181.

[KR20] Mihir Kale and Abhinav Rastogi. "Text-to-Text Pre-Training for Data-to-Text Tasks". In: *Proceedings of the 13th International Conference on Natural Language Generation*. Ed. by Brian Davis, Yvette Graham, John Kelleher, and Yaji Sripada. Dublin, Ireland: Association for Computational Linguistics, Dec. 2020, pp. 97–102.

[Lau+22] Hugo Laurençon, Lucile Saulnier, Thomas Wang, et al. *The BigScience ROOTS Corpus: A 1.6TB Composite Multilingual Dataset*. 2022. arXiv: 2303.03915 [cs.CL].

[Lew+19] Mike Lewis, Yinhan Liu, Naman Goyal, et al. *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. 2019. arXiv: 1910.13461 [cs.CL].

[LGA16] Rémi Lebret, David Grangier, and Michael Auli. "Neural Text Generation from Structured Data with Application to the Biography Domain". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Ed. by Jian Su, Kevin Duh, and Xavier Carreras. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1203–1213.

[LH17] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2017. arXiv: 1711.05101 [cs.LG].

[Lin04] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81.

[Liu+17] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. *Table-to-text Generation by Structure-aware Seq2seq Learning*. 2017. arXiv: 1711.09724 [cs.CL].

[Liu+21] Qian Liu, Bei Chen, Jiaqi Guo, et al. *TAPEX: Table Pre-training via Learning a Neural SQL Executor*. 2021. arXiv: 2107.07653 [cs.CL].

[Pap+02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318.

[Par+20] Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, et al. *ToTTo: A Controlled Table-To-Text Generation Dataset*. 2020. arXiv: 2004.14373 [cs.CL].

[Pos18] Matt Post. *A Call for Clarity in Reporting BLEU Scores*. 2018. arXiv: 1804.08771 [cs.CL].

[Raf+19] Colin Raffel, Noam Shazeer, Adam Roberts, et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2019. arXiv: 1910.10683 [cs.LG].

[Sán22] Sofía Sánchez. *Our new open-source model is here: From table to text with NARRATABLE*. Sept. 2022. URL: https://www.narrativa.com/our-new-open-source-model-is-here-from-table-to-text-with-narratable/.

[Sca+22] Teven Le Scao, Angela Fan, Christopher Akiki, et al. *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. 2022. arXiv: 2211.05100 [cs.CL].

[Tou+23] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL].

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[WSR17] Sam Wiseman, Stuart Shieber, and Alexander Rush. “Challenges in Data-to-Document Generation”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 2253–2263.

[Xu+23] Yiheng Xu, Hongjin Su, Chen Xing, et al. *Lemur: Harmonizing Natural Language and Code for Language Agents*. 2023. arXiv: 2310.06830 [cs.CL].

[Yen+23] Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, et al. *Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions*. 2023. arXiv: 2305.10435 [cs.CL].

[Zha+22] Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. “ReasTAP: Injecting Table Reasoning Skills During Pre-training via Synthetic Reasoning Examples”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 9006–9018.

[Zha+23] Yilun Zhao, Zhenting Qi, Linyong Nan, et al. *QTSumm: Query-Focused Summarization over Tabular Data*. 2023. arXiv: 2305.14303 [cs.CL].

[Zhe+23] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. 2023. arXiv: 2306.05685 [cs.CL].

[ZXS17] Victor Zhong, Caiming Xiong, and Richard Socher. *Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning*. 2017. arXiv: 1709.00103 [cs.CL].

List of Figures

List of Tables