Multilingual Table-to-Text Generation with Question-Answer Plans

Aden Haussmann

4th Year Project Report
Computer Science
School of Informatics
University of Edinburgh

2024

Abstract

Multilingual Natural Language Processing is challenging due to the lack of training data for low-resource languages. However, some of these languages have millions or tens of millions of speakers globally, making it important to improve NLP tools for them. Table-to-Text, the task of generating natural language descriptions of tables of data, is an excellent measure of models' reasoning abilities, but is very challenging in the multilingual setting. System outputs are often not attributable, or faithful, to the data in the source table. Intermediate planning techniques like Question-Answer (QA) blueprints have been shown to improve attributability on summarisation tasks. QA blueprints are concatenated question-answer pairs which relate to the input table, and are generated before generating the verbalisation itself, and they help control the content of the output. **This project aims to explore whether QA blueprints make multilingual Table-to-Text outputs more attributable to the input tables.** A challenging multilingual Table-to-Text dataset which includes African languages is extended with QA blueprints, which are generated and heuristically filtered. Sequence-to-sequence models (transformers) are then finetuned on this dataset, with and without blueprints. Two setups are tested; English, where the reference blueprint is in English and the reference verbalisation is in the target language, and translated, where the reference blueprint is also translated into the target language. Results show that blueprints improve performance for models finetuned and evaluated only on English, but do not demonstrate gains for multilingual models (with English blueprints performing significantly worse than translated ones). This is due to inaccuracies in machine translating the blueprints from English into target languages when generating the dataset to train on, and models' struggling to rely closely on the blueprints they generate. An in-depth analysis is conducted on why this is challenging.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Aden Haussmann*)

Acknowledgements

I would like to thank my outstanding supervisor, Dr. Mirella Lapata, for her guidance and support. To be supervised by such a distinguished researcher was a privilege.

I'd also like to thank Tom Sherborne, who endured my many technical questions, and helped me a great deal.

Finally, I would like to thank my parents, who gave me the opportunity to pursue my academic goals overseas.

Contents

Chapter 1

Introduction

1.1 Task: Table-to-Text Generation

Table-to-Text is a task in Natural Language Generation (NLG) that refers to taking a table of structured data which could represent a graph or chart etc., and generating natural language sentences, or verbalisations, that describe the data in the table. Table-to-Text can be considered a sub-task of the more general Data-to-Text, where the input is any form of structured data, such as JSON, database tables, knowledge graphs etc.

For a Language Model (LM) to be able to accept the table or chart as input, it is converted into a textual, or linearised, form. This retains the chart's data and structure entirely, but is now in a format that an LM can consume. This step is done by humans.

The Table-to-Text task is to generate **fluent** and **accurate** descriptions of the data in the input table. The LM takes the linearised table as input, and the goal is to output a verbalisation, like in Figure 1.1.

1.2 Motivation for Multilingual Table-to-Text Generation

Although incredible progress has been made in Natural Language Processing (NLP) in recent years, the majority of research, models and datasets focus on English (Ruder, 2022). Yet there are many languages with tens of millions of speakers which are severely underrepresented (low-resource languages, or LRLs). For example, Igbo is a language found predominantly in Nigeria that is considered low-resource despite being spoken by around 44 million people.[1] If NLP tools continue to rapidly improve for just a few languages, these other communities will be unable to use them, and be left behind. It is therefore important that more research is done to investigate techniques which make NLP work better for a more diverse set of languages.

So, why Table-to-Text? Table-to-Text is a highly challenging task for LMs and a very good way of evaluating their reasoning capability, as it often requires amalgamating data in multiple table cells and doing simple arithmetic.

[1] https://celt.indiana.edu/portal/Igbo/index.html

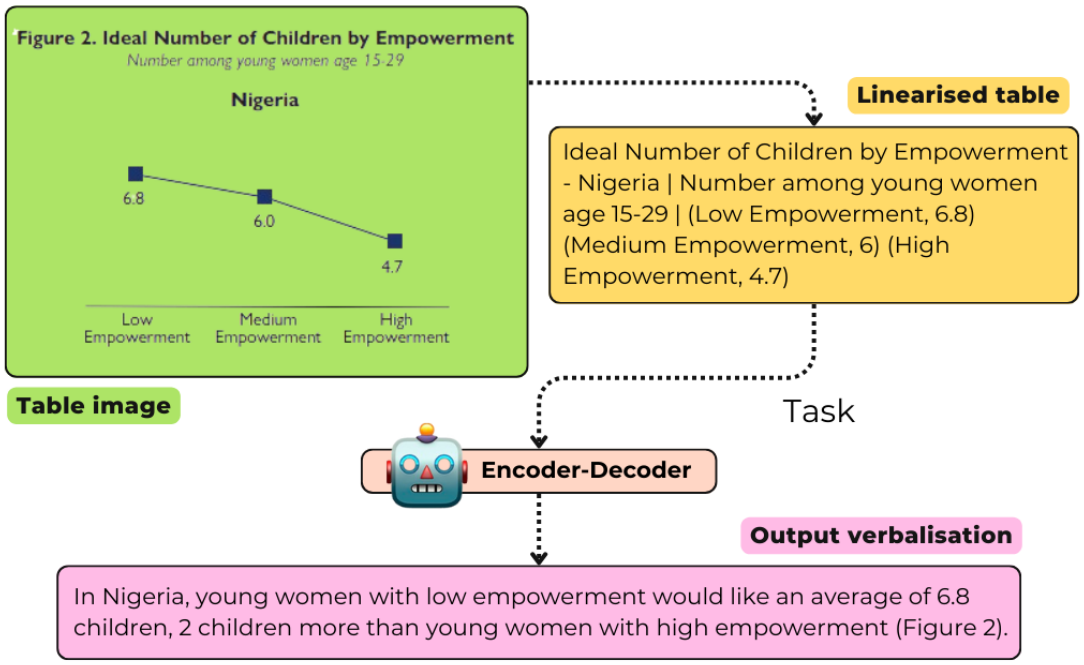


Figure 1.1: An example of a chart image, its linearised form (created by human annotators), and an output verbalisation (produced by a neural network).

Existing baseline multilingual models are somewhat capable of producing fluent verbalisations of tables in low-resource African languages, but these outputs are, more often than not, not attributable (faithful) to the input tables. In other words, they contain information that is not accurate to the data in the input table (Gehrmann et al., 2022). Thus this is an area with the potential for significant improvement.

1.3 Research Question

Intermediate planning refers to when models are trained to generate some planning text before the target output itself, instead of just the output. This will be introduced in detail in Section 2.2. This intermediate text serves as a content plan for the output. Intermediate planning techniques have been shown to improve faithfulness in tasks such as summarisation. These intermediate plans can, for example, take the form of Question-Answer pairs (Narayan et al., 2023) or entity chains (Narayan et al., 2021).

The goal of this project is to apply intermediate planning techniques that have been successful in other NLG tasks to the problem of multilingual Table-to-Text generation, and evaluate whether they affect the understandability (fluency), but in particular the attributability (faithfulness) of output verbalisations.

In summary, the following research question is explored:

To what extent do intermediate text plans, made up of Question-Answer pair blueprints, improve the attributability of multilingual Table-to-Text generation?

1.4 Key Contributions & Findings

Conditional generation with Question-Answer blueprints, a technique which has proven effective for improving the faithfulness of summaries, is applied to the task of Table-to-Text generation for the first time. Further, it is applied to a challenging multilingual Table-to-Text dataset containing African languages.

The results show that QA blueprints do increase attributability of outputs for models finetuned and evaluated on only English data. However, in the multilingual setting the technique is less effective. The challenge is twofold; inaccuracies in machine translating blueprints generated in English into target languages makes the training dataset less than perfect, baking in a fundamental disadvantage before training even begins. Furthermore, models struggle to produce verbalisations which rely heavily on their blueprints. This is analysed in detail in Chapter 7.

1.5 Report Structure

The rest of this report is laid out as follows. Chapter 2, *Background*, introduces the task, the neural models used, low-resource languages, and the challenges of evaluation in more detail. Chapter 3, *Data preprocessing*, gives an overview of how the dataset is prepared. Chapter 4, *QA blueprint creation*, describes the process of generating and heuristically filtering Question-Answer pairs to build blueprints from. Chapter 5, *Model finetuning and experimental setup*, gives details on how models are trained and tested. Chapter 6, *Automatic evaluation*, reviews various metrics' suitability for evaluation on the task, and justifies metric choices. Chapter 7, *Results*, reports and analyses model results in detail. Finally Chapter 7, *Conclusion*, offers a summary of the findings and recommendations for future work and improvements.

Chapter 2

Background

2.1 Transformers & Neural Language Models

2.1.1 Sequence-to-Sequence Models

In 2014, sequence-to-sequence (seq2seq) learning was proposed, i.e. using an encoder-decoder neural network to map an input sequence to an output sequence. This architecture is ideal for applications such as machine translation or summarisation (Sutskever et al., 2014). Crucially, the encoder is not limited to text as input, but can encode arbitrary sequences. This includes structured representations such as tables or images. The seq2seq framework's flexibility has made it the standard for NLG (Ruder, 2018). Encoders and decoders were typically RNNs or sometimes LSTMs, but in recent years, the relatively new transformer architecture has become highly popular.

The proposal of attention (Bahdanau et al., 2016), specifically self-attention (Cheng et al., 2016), was one of the key insights which led to the development of a new neural architecture, the transformer (Vaswani et al., 2017). The transformer is a simplified architecture based purely on attention, that does away with recurrence and convolutions completely. It was shown to be more parallelisable and faster to train, and achieved a new state-of-the-art performance on machine translation tasks.

A transformer-based encoder-decoder defines a conditional distribution of target vectors $\mathbf{Y}_{1:m}$ given an input sequence $\mathbf{X}_{1:n}$, where m and n are the lengths of the output and input sequences respectively:

$$p_{\theta_{enc},\theta_{dec}}(\mathbf{Y}_{1:m} \mid \mathbf{X}_{1:n}) \tag{2.1}$$

Given an input sequence $\mathbf{X}_{1:n}$, a transformer-based encoder maps this to a sequence of hidden states, $\overline{\mathbf{X}}_{1:n}$:

$$f_{\theta_{enc}} : \mathbf{X}_{1:n} \rightarrow \overline{\mathbf{X}}_{1:n} \tag{2.2}$$

The decoder models a conditional distribution of the target sequence $\mathbf{Y}_{1:m}$ given the encoded hidden states $\overline{\mathbf{X}}_{1:n}$:

$$p_{\theta_{dec}}(\mathbf{y}_i \mid \mathbf{Y}_{1:m}, \overline{\mathbf{X}}_{1:n}) \tag{2.3}$$

By Bayes' rule, this can be factorised to yield a conditional distribution of the target vector $\mathbf{y}_i$ given the encoded hidden states $\overline{\mathbf{X}}_{1:n}$ and all previous target vectors $\mathbf{Y}_{0:i\text{-}1}$:

$$p_{\theta_{dec}}(\mathbf{y}_i \mid \mathbf{Y}_{1:m}, \overline{\mathbf{X}}_{1:n}) = \prod_{i=1}^{m} p_{\theta_{dec}}(\mathbf{y}_i \mid \mathbf{Y}_{0:i\text{-}1}, \overline{\mathbf{X}}_{1:n}) \tag{2.4}$$

Here, the decoder maps the encoded hidden states, and all previous target vectors $\mathbf{Y}_{0:i\text{-}1}$, to the logit vector $\mathbf{l}_i$. The *softmax* function is applied to the logit vector to produce the conditional distribution $p_{\theta_{dec}}(\mathbf{y}_i \mid \mathbf{Y}_{0:i\text{-}1}, \overline{\mathbf{X}}_{1:n})$.[1]

The distribution of the target vector $\mathbf{y}_i$ is **explicitly conditioned on all previous target vectors**. The output is *auto-regressively* generated from this distribution.

So, at each step, when the decoder generates the next token, it is conditioned on the encoder output, i.e. the encoded representation of the input sequence. But it is also conditioned on all previous decoder outputs. The decoder attends to all previously generated tokens to maintain context, helping make outputs consistent and coherent.

2.1.2 T5

T5, or *"Text-to-Text Transfer Transformer"* (Roberts et al., 2019), is a sequence-to-sequence transformer-based model with an encoder-decoder architecture. T5 is capable of performing typical sequence-to-sequence tasks such as summarisation, but is also trained to do classification and text-to-text regression.

T5 employs Transfer learning, which is the process of pretraining a model on some task for which there is a huge quantity of high-quality data, before finetuning the model on some other specific downstream tasks.

T5 is pretrained on the open-source C4, or *"Colossal Clean Crawled Corpus"*, dataset[2], which is based on a single month's worth of scraped web data in the Common Crawl (CC) dataset, but applies several filtering heuristics to create a subset of high-quality data, removing everything which is not natural language. C4 is 750GB in size.

During pretraining, words or spans in the input are masked, and the model is trained to predict these. For example, in Figure 2.1, in the two-word span "for inviting", the two words are not predicted separately, but as a single span. This allows T5 to capture structure in language and gives it an advantage over single-token masking.

[1] Equations are taken from https://huggingface.co/blog/encoder-decoder#encoder-decoder, although I believe there is a mistake in the notation and have opened a PR to fix it: https://github.com/huggingface/blog/pull/1942.

[2] https://www.tensorflow.org/datasets/catalog/c4

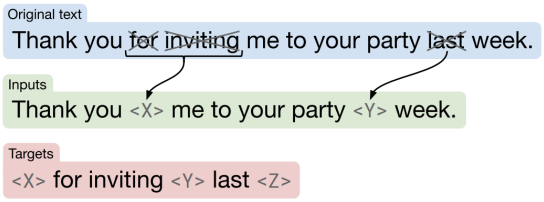


Figure 2.1: Schematic of the objective used in pretraining the T5 baseline model (Roberts et al., 2019), showing the masking and placeholders.

Downstream performance is studied on tasks including machine translation, question answering, abstractive summarisation and text classification. This ability to perform several tasks is made possible by prefixing the input with a token which indicates which task the model should do, as shown in Figure 2.2. This figure also clearly shows that all tasks, even those with numerical outputs, are formulated as text-to-text.

When released by Google in 2019, T5 achieved state of the art (SOTA) performance on several benchmarks.

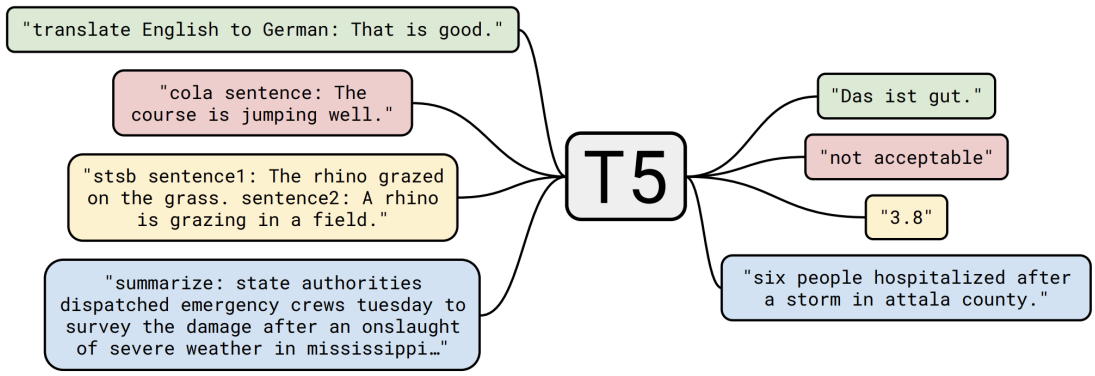


Figure 2.2: T5's text-to-text framework (Roberts et al., 2019).

2.1.3 mT5

Transfer learning is a popular and effective method for multilingual NLP (Magueresse et al., 2020). mT5, or *"Massively Multilingual pre-trained Text-to-Text Transformer"* (Xue et al., 2021) is a multilingual version of the T5 model, and is the model that is used in this project.

Its architecture and training methods are very similar to that of T5, with a few differences. One such difference is that mT5 uses Gated Exponential Linear Units instead of regular Gated Linear Units, as the GeGLU activation function was found to be more effective than GLU, which can be susceptible to overfitting and vanishing gradients in large models. Another important difference is in the sampling techniques used to mitigate the difference in the amount of training data available for high and low-resource languages. The sampling probability is inversely proportional to the square root of the number of available examples for a given language, $p(L) \propto p\,|L|^{\alpha}$. Finally, mT5's vocabulary is increased from 32,000 to 250,000 sub-word units to represent 101 languages. The downside of this is that the model requires more memory and compute resources to train.

mT5 is pretraied on the mC4 dataset[3], a multilingual version of C4. mC4 is based on 71 months of Common Crawl data instead of one, to gather a greater diversity of languages. cld3, Google's *"Compact Language Detector"*[4] is used to identify the languages.

Unlike T5, mT5 does not undergo task-specific training with prefixes and therefore cannot be used without additional finetuning.

mT5 is released in 5 checkpoints, with these numbers of parameters: Small (300M), Base (580M), Large (1.2B), XL (3.7B) and XXL (13B).

2.2 Faithfulness in Natural Language Generation

Despite the significant recent advances in neural generation models, they still have shortcomings. These include tendencies to hallucinate and repeat themselves, and they struggle to remain faithful to input data and identify important information. Solving these problems is challenging because deep neural networks are opaque by nature, making it difficult to understand their reasoning and find the root of errors (Narayan et al., 2023).

Numerous techniques have been applied in attempts to reduce hallucination and make outputs more faithful and attributable to the source document (Narayan et al., 2023). One family of such techniques is centred around content selection and planning, whereby the model is trained to identify and extract relevant information from the input, and generate an "intermediate plan" of *what to say* (selection) and *in what order* (planning) before generating the actual output, which is conditioned on the plan.

Why do intermediate plans work? Recall from Equation 2.4 that at each step, when a transformer-based decoder generates the next token, that token is conditioned not only on the encoder output, i.e. the encoded representation of the input sequence, but also on previous decoder outputs. It is this mechanism that means when the decoder begins generating the verbalisation, having already generated the blueprint, the verbalisation will be conditioned on the blueprint as well as the input sequence.

One approach to creating intermediate plans for abstractive summarisation is entity chaining (Narayan et al., 2021). This involves creating an ordered set of entities from the target summary and prepending it to the summary. The model is trained to generate the entity chain, then continue generating the summary itself, which is conditioned on the chain and the input document. The chief benefit of this approach is that it is very simple but effective.

Suppose d is an input document. The model is trained to generate the content plan c for summary s as $p(c|d)$, then the summary s as $p(s|c,d)$. So, the model encodes document d and generates the concatenated plan and summary $c;\ s$. c and s are prefixed with special tokens. $p(s|c,d)$ (Narayan et al., 2021).

A similar idea is to build the intermediate plans from question and answer (QA) pairs instead of entity chains, where the questions and answers are generated from the target

[3] https://www.tensorflow.org/datasets/community_catalog/huggingface/mc4
[4] https://github.com/google/cld3

summaries. This has been shown to allow more control of a model's output, and more explanation of a model's features than entity chains (Narayan et al., 2023). The approach is much the same. The difference is the plan c is not an entity chain, but a concatenated set of question-answer (QA) pairs $a_1;q_1;a_2;q_2;\ldots;a_n;q_n$. See Figure 2.3 for an example of questions and answers generated from a summary.

This method was inspired by the Questions Under Discussion (QUD) model of discourse. QUD, a tool used by linguists and language philosophers, asserts that the structure of a text can be expressed by extracting questions that are raised in the text. The theory is that for each assertion made, the text contains implicit questions, to which the answer is the assertion (Benz and Jasinskaja, 2017). Recent research has made progress towards automating the generation of all salient questions for a sentence (De Kuthy et al., 2020).

Q_1: Who built the Shelby Mustang from 1969 to 1970?	A_1: Ford
Q_2: During what years was the Shelby Mustang built by Shelby American?	A_2: 1965 to 1968
Q_3: In what year was the fifth generation of the Ford Mustang introduced?	A_3: 2005
Q_4: What was the Shelby Mustang revived as?	A_4: a new high-performance model

The Shelby Mustang is a high performance variant of the Ford Mustang which was built by Shelby American from 1965 to 1968, and from 1969 to 1970 by Ford. Following the introduction of the fifth generation Ford Mustang in 2005, the Shelby nameplate was revived as a new high-performance model, this time designed and built by Ford.

Figure 2.3: Example of questions and answers generated from a summary (Narayan et al., 2023).

Content selection and planning has also proved effective in improving Table-to-Text outputs. A "plan" of *what to say* and *in what order* is generated, before generating the verbalisation itself. The neural model then uses this plan while generating the verbalisation (Puduppully et al., 2019). In 2019 such a model achieved state-of-the-art BLEU scores on the RotoWire[5] (Wiseman et al., 2017b) and ToTTo (Parikh et al., 2020)[6] datasets. This work used an LSTM-based encoder-decoder architecture with an attention mechanism and has since been surpassed by simply finetuning $T5_{XL}$ (Kale and Rastogi, 2020). This showed that the modern text-to-text pretraining-finetuning paradigm employed by models such as mT5 works very well for Table-to-Text tasks.

2.3 The Challenge of Evaluation

Just as achieving faithfulness to the source table is a challenge in Table-to-Text generation, so too is evaluating whether a text is understandable and attributable.

Although human evaluators generally provide the best judgements for NLG systems, the process of designing and running experiments is expensive, time-consuming, nuanced and can require ethics approval. This poses a challenge to rapid model development and research. Therefore, automated evaluation metrics which act as a proxy for human evaluations and are cheap to compute, are critical (Sellam et al., 2020).

Two of the most frequently-used such metrics are BLEU and ROUGE. Introduced in 2002, BLEU (Bilingual Evaluation Understudy) is a metric for evaluating machine

[5] https://paperswithcode.com/sota/data-to-text-generation-on-rotowire
[6] https://paperswithcode.com/sota/data-to-text-generation-on-totto

translations (Papineni et al., 2002). In 2004, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) was proposed as a metric for evaluating summaries. Both use Ngram overlap (Where Ngrams are series of N adjacent tokens; letters, parts of words, or words). to measure similarity between a prediction and a set of references, and have proved popular for evaluating Table-to-Text generation over the past two decades. BLEU is used in this project, and explored further in Section 6.1.

However, BLEU and ROUGE have been repeatedly shown to correlate poorly with human evaluations (Sellam et al., 2020). In 2015, CHRF, a character Ngram metric also intended for evaluating machine translation, was developed. It showed comparable or superior correlations with human judgement to BLEU (Popović, 2015), so it is also used in this project Section 6.1.

Still, these metrics are not ideal for evaluating Table-to-Text generation. It has been observed that BLEU rewards fluent outputs, but not those which accurately reproduce the data in the table, making it particularly inadequate (Wiseman et al., 2017a).

Furthermore, they all rely on the assumption that the reference is an optimal target, or *gold*. In practice, because datasets are collected heuristically and automatically, this is often not true (Dhingra et al., 2019). To address this, PARENT (Precision And Recall of Entailed N-grams from the Table) was proposed in 2019. PARENT is a metric for evaluating Table-to-Text generation that combines the reference and table when calculating precision. This rewards correct data in the output which is present in the table but not the reference (Dhingra et al., 2019).

Other metrics have been developed with factuality, rather than just fluency, in mind. FACTKB (Feng et al., 2023) is a learned metric intended to evaluate the factuality of summarisations. It evaluates summaries against source documents instead of references, and returns a measure of how factual the summary is. FACTKB performs well in out-of-domain settings, therefore this project shall consider whether it can be used to evaluate Table-to-Text outputs (Section 6.2).

The faults of automatic metrics are considered to impede the recent progress being made by neural models on NLG tasks (Sellam et al., 2020). Despite the apparent continuous advancement of automated evaluation through the proposal of new and improved metrics, particularly learned metrics, most are all still insufficient to accurately assess performance on challenging multilingual Table-to-Text tasks.

2.4 Multilingual NLG and Low-Resource Languages

Modern neural models need to be pretrained on huge amounts of data. For example, GPT-3 was trained on 570GB of text, filtered from 45TB of compressed plain text sourced from crawling the internet, Wikipedia and books (Brown et al., 2020). In such sources, widely-spoken languages such as English, French or German will be well-represented as they enjoy large populations of speakers and many years of books and web articles being authored. However, languages with less written content - especially online - will not be prevalent in such crawls. This means models are less likely to have seen these languages before and can therefore not be used for downstream tasks in these

languages. These are called low-resource languages (LRLs).

A review of past work shows that there are several ways to improve the fundamental lack of data for LRLs. One is by expanding and augmenting datasets, either through neural and computational methods on existing datasets, or of course by manually creating new datasets (Magueresse et al., 2020).

Most Table-to-Text datasets - like the popular ToTTo (Parikh et al., 2020) - are in English, making multilingual Table-to-Text generation challenging. This is addressed by TATA: *"A Multilingual Table-to-Text Dataset for African languages"* (Gehrmann et al., 2022). TATA is a fully parallel dataset of Table-to-Text samples in 9 languages, including four African languages (Hausa, Igbo, Swahili, and Yorùbá) and Russian as a zero-shot test language (A language to test the model on that is not represented in the training data, used to evaluate how well the model will generalise to new languages). At 8,700 samples, TATA is relatively small (ToTTo has 120,761 training samples).

A key difference between ToTTo and TATA is that ToTTo tables have highlighted cells which control what information the verbalisation should include. TATA does not have this, making the problem much less constrained. For any table in TATA, there are far more valid verbalisations. This might make achieving high scores on TATA fundamentally more difficult. Examples of input/output pairs from TATA can be seen in Table 2.1.

Setting	nU – U – U+A	Reasoning	# Cells
Reference		0.40 / 0.75	$8.0_{6.7}$
$mT5_{small}$		0.03 / 0.78	$6.9_{5.9}$
$mT5_{SSA}$		0.03 / 0.77	$6.8_{5.1}$
$mT5_{XXL}$		0.34 / 0.77	$7.9_{6.0}$

Figure 2.4: Human evaluation of TATA, where nU (red) means not understandable, U (grey) means understandable and U+A (green) means understandable and attributable. The #Cells column represents the number of cells reasoned over, with std. dev. The references and $mT5_{XXL}$ have a U+A rate of 0.53 and 0.44 respectively (Gehrmann et al., 2022).

mT5, finetuned on TATA, achieves poor understandability and attributability as can be seen in Figure 2.4. Even $mT5_{XXL}$ could only achieve understandability and attributability for 44% of its outputs, 9% below the reference rate.

Perhaps as expected, mT5's BLEURT, ROUGE and CHRF scores on TATA are very low and do not correlate well with human evaluations. Thus, these were deemed inadequate to assess model performance on the dataset and the learned STATA, *"Statistical Assessment of Table-to-Text in African languages"*, metric was created by finetuning mT5 on the human evaluations of model outputs on the validation set and references. STATA has a far higher correlation with human annotations. This is explored further in Section 6.3.

This reinforces the proposition that commonly-used automatic metrics are not accurate judges of outputs on challenging datasets such as TATA, especially where attributability is being evaluated.

In the above baseline model, mT5 is simply taking as input, the linearised tables,

and as references, target verbalisations which it learns to generate. There are various intermediate planning techniques for content planning and selection that have been applied to the task of summarisation, as discussed in Section 2.2, which could be applied here to improve model performance. This shall be the focus of the project, and STATA (Section 6.3), in combination with the other automatic metrics introduced, shall be used for evaluation.

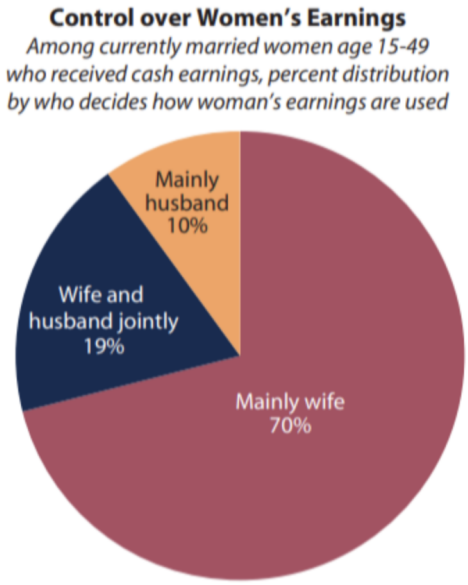


Figure 2.5: An example of a table whose linearised form appears in TATA (Table 2.1).

Lang	Linearised input table	Target verbalisation
English	Control over Women's Earnings | Among currently married women age 15-49 who received cash earnings, percent distribution by who decides how woman's earnings are used | (Mainly wife, 0.7) (Wife and husband jointly, 0.19) (Mainly husband, 0.1)	One in five married working women made joint decisions with their husbands on spending cash earnings while 10% report that their husbands make the decision alone.
Swahili	Udhibiti juu ya Mapato ya Wanawake | Miongoni mwa wanawake walio na umri wa miaka 15-49 ambao wameolewa kwa sasa waliopokea mapato ya pesa, asilimia ya usambazaji wa anayeamua jinsi ambavyo mapato ya mwanamke yatatumika | (Mke hasa, 0.7) (Mke na mume kwa pamoja, 0.19) (Mume hasa, 0.1)	Mmoja kati ya wanawake watano wanaofanya kazi na wameolewa walifanya maamuzi ya pamoja na mume zao kuhusu kutumia mapato yao huku 10% waliripoti kuwa mume zao walifanya maamuzi peke yao.
Hausa	Kula da Kudin da Mata ke Samu | A tsakanin matan da ke da aure a yanzu haka 'yan shekaru 15-49 wadanda ke karbar kudi tsaba, kason rabe-raben na wanda ke yanke hukunci a kan yadda za a yi amfanida kudin da mata ke samu | (Mafi yawa mata ce, 0.7) (Mata da miji a hade, 0.19) (Mafi yawa miji ne, 0.1)	Daya daga cikin matan aure biyar da ke aiki sukan yanke hukunci tare da mazansu a kan yadda za a kashe kudin da suke samu a yayin da rahoto ya bayyana cewa, kashi 10%, mazansu ne ke yanke hukunci su kadai.

Table 2.1: Some parallel examples of Figure 2.5 in TATA in different languages.

Chapter 3

Data Preprocessing

3.1 Building the Datasets

TATA contains some samples without references. Gehrmann et al. (2022) experiment with different ways of handling this. The first is to simply skip these references and not train on them. The second is a tagging system where "0" is appended to inputs with empty references and the model learns to predict an empty string, and "1" is appended to inputs with references. Since Gehrmann et al. (2022) found that the first approach performed better than the second, only the first approach is done in this project. [1]

TATA has the following columns: example_id, title, unit_of_measure, chart_type, was_translated, table_data, linearized_input and table_text.

The *table_text* column contains a list of all references. To increase the amount of training data, the training set is exploded along this column, repeating the sample for every reference in the list. After cleaning and exploding references, the training set has 7,060 rows, the validation set has 754 and the test set has 763.

For the validation set, the first reference in the *table_text* column is chosen as the target. At test time, metrics are computed between the prediction and each of the references in *table_text*, and the highest score is taken.

A separate dataset, which contains only the English examples from the main set, is also created to finetune English-only models. This set's test, validation and test splits have 902, 100 and 100 rows respectively.

3.2 Bugs in the Dataset

While preprocessing the data, it was noticed that several rows contained errors in the *table_text* column. 12 rows had some references which matched the data, then several commas in a row (effectively empty references), followed by "START OF TEXT" which is an instruction for human annotators that wasn't meant to make it into the dataset, and

[1] So, all results in this project are derived from a setup equivalent to SKIP NO REFERENCES from Gehrmann et al. (2022).

finally some more references which were totally unrelated to the data. There were also some more rows which just had repeating commas.

At least some of the unrelated references appeared elsewhere in the dataset. For example, *"Four percent of Tanzanian women age 15-49 reported having two or more sexual partners in the past 12 months."* appeared in example *DM51-en-3*, where it should not have, and also in *SR196-en-3*, where it belonged.

To solve this, the commas, annotator tags and incorrect references are deleted from the affected rows, and the amended rows are left in the dataset so as not to lose training data. It is assumed that references after the "START OF TEXT" instruction do not belong there as this is what was observed for the English examples, but this is not verified for the non-English examples.

The merged pull request with the fix in Google's official TATA repository can be seen here: https://github.com/google-research/url-nlp/pull/6.

Chapter 4

QA Blueprint Creation

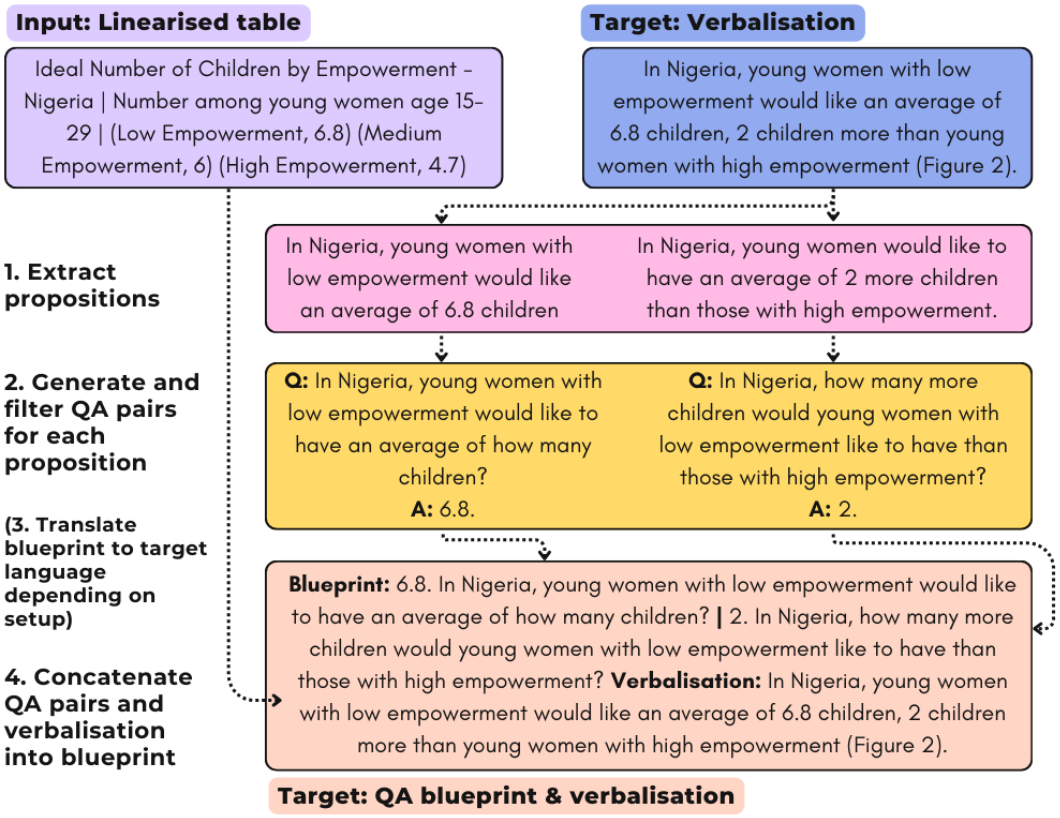


Figure 4.1: The process of generating QA blueprints.

4.1 Generating & Heuristically Selecting QA Pairs

A similar process for creating the QA blueprints is followed to that of Narayan et al. (2023), but with some differences.

The following is done for each English reference (which are single sentences):

First, propositions are extracted from the reference. A proposition is a sub-sentential unit of logic, or can be thought of as a fact contained within the sentence. Narayan et al. (2023) break sentences into propositions by splitting on punctuation, prepositions, relative pronouns and coordination. In this project, a different approach is taken. FLAN-T5-Large (Chung et al., 2022) finetuned[1] for "propositionizing" sentences by Chen et al. (2023) is used generate a minimal sentence for each proposition in the reference.

For example, from the sentence from Figure 4.1:

> *In Nigeria, young women with low empowerment would like an average of 6.8 children, 2 children more than young women with high empowerment (Figure 2).*

the following propositions are extracted:

- *In Nigeria, young women with low empowerment would like a average of 6.8 children.*
- *In Nigeria, young women with low empowerment would like to have an average of 2 more children than those with high empowerment.*

The advantage of this approach is that the additional step of matching QA pairs generated on the whole reference to a single proposition can be skipped. Instead, the QA pairs are generated directly from each proposition, instead of the reference itself. This guarantees that there will not be overlapping or redundant information, i.e. each QA pair expresses a different fact (assuming there is no information overlap between propositions).

Next, for each proposition, 5 QA pairs are generated using T5-Large finetuned[2] on SQuAD (Rajpurkar et al., 2016) for QA generation by Manakul et al. (2023). 5 QA pairs are generated because trial and error showed this to be sufficient to get some high-quality pairs. These are generated with *do_sample* = *True*[3] to get different outputs.

Next, QA pairs where the question does not end with a question mark, or the question or answer are just empty strings, are dropped. This cleans up the QA pairs and removes anomalous generations which do occur occasionally.

A regex is used to identify numbers in the remaining QA pairs, and pairs containing a hallucinated number, that is not present in the source reference, are dropped. The goal of the QA blueprints is to focus the model's output on factual data which is attributable to the input table. If numbers are hallucinated at this point, this error will propagate into the final output verbalisation and defeat the blueprint's purpose.

QA pairs where the answer is fully contained within the question are dropped.

For any duplicated answers, the QA pair where the question has the greatest lexical similarity with the proposition (calculated via a word-level F1 score between the question and reference), is chosen. The other is dropped.

[1] https://huggingface.co/chentong00/propositionizer-wiki-flan-t5-large
[2] https://huggingface.co/potsawee/t5-large-generation-squad-QuestionAnswer
[3] https://huggingface.co/docs/transformers/generation_strategies

Finally, the QA pair with the greatest lexical similarity (the highest number of overlapping words) with the reference is selected. For each of the propositions in the above example respectively, these are:

- *"Question: In Nigeria, young women with low empowerment would like an average of how many children? Answer: 6.8."*
- *"Question: In Nigeria, how many more children would young women with low empowerment like to have than those with high empowerment? Answer: 2."*

Narayan et al. (2023) filter QA pairs by dropping ones where the answer does not appear at the end of the proposition. They do this because of the theme-rheme structure of sentences in natural language (Culicover and McNally, 2020) where known information, the theme, usually comes first and new information, the rheme, usually comes after (Kruijff-Korbayová and Steedman, 2003). Therefore, dropping QA pairs where the answer does not appear at the end of the proposition selects for questions that focus on new information.

In this project, this step is skipped. This is because for a proposition like:

> *"In Nigeria, young women with low empowerment would like to have an average of 2 more children than those with high empowerment."*

the QA pair

> *"Question: In Nigeria, who would young women with low empowerment like to have an average of 2 more children than? Answer: Young women with high empowerment."*

will be favoured over

> *"Question: In Nigeria, how many more children would young women with low empowerment like to have than those with high empowerment? Answer: 2."*

but this is arguably not preferable. The second QA pair would be dropped, but it is more natural and focuses on the numerical figure, which is the most important item in the proposition.

On average, a QA blueprint generated via this strategy contains two or three QA pairs.

4.2 Building the Blueprint

The chosen QA pairs are then concatenated as follows: Let a denote an answer, and q denote a question. The blueprint b takes the form $a_1;q_1;...;a_n;q_n$. Although it is less natural, answers come before questions because Narayan et al. (2023) found it yielded better results. Answers and questions are separated by a full stop ".", and QA pairs are separated by a pipe "|".

Special tokens *"Blueprint:"* and *"Verbalisation:"* are prefixed to the blueprint b and verbalisation v respectively, and these are concatenated to form the references.

Narayan et al. (2023) call this a "global" blueprint and dub it an End-to-End model, as it determines what content to focus on for the entire output. They experiment with two more types of planning: The Multi-Task model is trained to do two distinct tasks, generate the concatenated answer plan and output, and answer plan with questions. This means the model does not have to generate such long outputs. The Iterative model interleaves planning with generation instead of creating a global plan before generating. It is trained to iteratively plan and generate one sentence at a time. The Iterative and Multi-Task plans mitigate the fact that generally, encoder-decoder models struggle to generate long texts, but this project focuses on the E2E plan since the verbalisations considered here are typically shorter than summaries as they are only one sentence.

In this setup, the encoder-decoder model takes the linearised table t as input and learns to predict the blueprint b as $p(b|t)$, then generate the output v as $p(v|t,b)$. It is noted by Narayan et al. (2023) that this relies on the blueprints being correct, and acknowledged that errors at this stage will propagate down the pipeline and affect the final generated verbalisation.

A limitation of generating QA pairs from the reference verbalisations is that the blueprints in the training set will never include data that do not appear in the reference verbalisations. More comprehensive QA pairs cannot be generated directly from TATA's documents, as these are linearised tables[4], so they are generated from the reference verbalisations, which are not guaranteed to mention all the data in the chart.

4.3 Making Blueprints Multilingual

The setup as described above would be sufficient if the goal was only to produce English generations. However, the goal is to produce multilingual generations. Thus, two blueprint setups are created. The first just uses the English blueprints. So, when the model generates output v as $p(v|t,b)$, b is always in English and v is in the target language. The second still generates the blueprints in English and translates them to the target language. So, both b and v are in the target language. For the English blueprint dataset only, a language tag, for example *"(Yorùbá)"* is also inserted after *"Verbalisation"* to help indicate to the model what the target language for the verbalisation is.

Machine translating (MT) QA pairs from English to the other languages will propagate small errors in translation further down the pipeline and hinder the effectiveness of the blueprints. To evaluate this risk, an analysis of translation quality is done on the training set. Since the dataset is parallel (i.e., each sample appears in every language), there is ample data to test the quality of machine translations in this specific context.

Each English sample's reference is translated, using Google Translate,[5] into the seven other languages which appear in the training set, and these translations are compared to the corresponding samples written in the target languages with automatic MT metrics. The results are recorded in Table 4.1.

[4] QA pair generation from tabular data would be an extremely interesting and useful project in its own right.

[5] https://pypi.org/project/googletrans/

	Hausa	Igbo	Swahili	Yorùbá	French	Arabic	Portuguese
CHRF	0.53	0.56	0.68	0.19	0.80	0.57	0.66
BLEU	0.30	0.37	0.47	0.06	0.68	0.33	0.47

Table 4.1: CHRF and BLEU scores for Google Translate-powered translations of English samples in TATA compared to corresponding samples in the target languages.

Table 4.1 shows that translation quality for the more broadly spoken languages such as French and Portuguese is high, with lower quality for the African languages. This is to be expected, as mT5 will have seen more English and French samples during its training than, say, Igbo. Swahili enjoys the best translations of the African languages, achieving higher scores than even Arabic. All languages can be translated with reasonable quality, with the exception of Yorùbá, which achieves the lowest scores by a large margin. It is not clear why Yorùbá scores so lowly, but there is discussion online indicating that others have noted Google Translate's poor Yorùbá quality.[6]

4.4 Examples

The blueprint, generated in English (see Table 4.2 for full examples), is:

> *13%. How much of young women in Mali are in the highest tercile for empowerment? |81%. What is the percentage of young women in the Philippines who are in the highest tercile for empowerment? |*

This is translated into Swahili as:

> *13%. Je, ni idadi gani ya wanawake vijana nchini Mali walio katika eneo la juu zaidi la kuwezeshwa? | 81%. Je, ni asilimia ngapi ya wanawake vijana nchini Ufilipino walio katika eneo la juu zaidi la kuwezeshwa? |*

Translated back into English (again, with Google Translate) to check the quality:

> *13%. What is the proportion of young women in Mali in the highest area of empowerment? | 81%. What percentage of young women in the Philippines are in the highest empowerment zone? |*

This is relatively close to the original English blueprint, however it loses the term "tercile" and replaces it with "area" or "zone", which could have many different meanings (e.g. spatial) that do not closely relate to "tercile". Swahili is one of the languages with better translation performance. For those with even poorer translations, the blueprints will not match perfectly to the verbalisations, which are translated by expert humans (Gehrmann et al., 2022).

[6] https://support.google.com/translate/thread/241706891/why-is-google-translate-translation-for-yoruba-so-bad?hl=en

Dataset	Source verbalisation	Generated blueprint	New reference
English	Only 13% of young women in Mali are in the highest tercile for empowerment compared with 81% of young women in the Philippines.	13%. How much of young women in Mali are in the highest tercile for empowerment? |81%. What is the percentage of young women in the Philippines who are in the highest tercile for empowerment? |	Blueprint: 13%. How much of young women in Mali are in the highest tercile for empowerment? |81%. What is the percentage of young women in the Philippines who are in the highest tercile for empowerment? |Verbalisation: Only 13% of young women in Mali are in the highest tercile for empowerment compared with 81% of young women in the Philippines.
Translated blueprints	Asilimia 13 pekee ya wanawake wadogo katika Mali ndio wapo katika kikundi cha juu cha uwezeshaji ikilinganishwa na asilimia 81 ya wanawake wadogo katika Ufilipino.	13%. Je, ni idadi gani ya wanawake vijana nchini Mali walio katika eneo la juu zaidi la kuwezeshwa? |81%. Je, ni asilimia ngapi ya wanawake vijana nchini Ufilipino walio katika eneo la juu zaidi la kuwezeshwa? |	Blueprint: 13%. Je, ni idadi gani ya wanawake vijana nchini Mali walio katika eneo la juu zaidi la kuwezeshwa? |81%. Je, ni asilimia ngapi ya wanawake vijana nchini Ufilipino walio katika eneo la juu zaidi la kuwezeshwa? |Verbalisation: Asilimia 13 pekee ya wanawake wadogo katika Mali ndio wapo katika kikundi cha juu cha uwezeshaji ikilinganishwa na asilimia 81 ya wanawake wadogo katika Ufilipino.
English blueprints	Seules 13% des jeunes femmes au Mali se situent dans le tercile le plus élevé en matière d'autonomisation contre 81% des jeunes femmes aux Philippines.	13%. How much of young women in Mali are in the highest tercile for empowerment? |81%. What is the percentage of young women in the Philippines who are in the highest tercile for empowerment? |	Blueprint: 13%. How much of young women in Mali are in the highest tercile for empowerment? |81%. What is the percentage of young women in the Philippines who are in the highest tercile for empowerment? |Verbalisation (French): Seules 13% des jeunes femmes au Mali se situent dans le tercile le plus élevé en matière d'autonomisation, contre 81% des jeunes femmes aux Philippines .

Table 4.2: Example blueprints from the three dataset setups for the same input table with English, Swahili and French as target languages.

Chapter 5

Experimental setup

5.1 Training Details

Gehrmann et al. (2022) use $\text{mT5}_{\text{Small}}$ and mT5_{XXL}. However, mT5_{XXL}, and indeed mT5_{XL} are both massive models (13B and 3.7B parameters respectively). Available compute resources during the completion of this project made finetuning these models infeasible. The largest model that could be trained is $\text{mT5}_{\text{Large}}$ (1.2B), so this is used instead.[1]

Both $\text{mT5}_{\text{Small}}$ and $\text{mT5}_{\text{Large}}$ are finetuned, with a conditional generation head[2], with the following hyperparameters and setup for all experiments:

- Constant learning rate of 0.001.
- Dropout of 0.1.
- Per-device batch size of 4.
- 5 epochs for the Small model and 3 for Large.
- Weight decay of 0.001 for the Large model.
- Linearised tables (inputs) and references are both truncated to 512 tokens. This figure is chosen based on an analysis of lengths of un-truncated tokenised samples, as seen in Table 5.1. There are a few inputs which are very long, however most are under 512 tokens in length.
- Validation loss is measured every 100 or 500 steps depending on batch size and after training the checkpoint with the lowest loss is selected.
- Training is done on a single NVIDIA A40 GPU.

Training curves for all models are plotted in Appendix A.

[1] Future expansion of this project could explore using Low-Rank Adaptation (LoRA) to finetune mT5_{XXL} with less GPU compute (Hu et al., 2021).

[2] https://huggingface.co/docs/transformers/model_doc/mt5#transformers.MT5ForConditionalGeneration

Split	Input/Reference	Mean	Median	Range (Min - Max)
Train (no Blueprints)	Input	178.66	139.5	10 - 2266
	Reference	60.35	52	9 - 371
Train (with Blueprints)	Input	178.96	140	10 - 2266
	Reference	122.78	114	31 - 545
Validation	Input	207.98	166.5	35 - 1210
	Reference	52.97	47	11 - 214
Test	Input	211.51	161	31 - 1808
	Reference	54.94	49	14 - 194

Table 5.1: Lengths of tokenised inputs (linearised table texts) and references (verbalisations) for the train, validation and test datasets.

5.2 Repetition Penalty

It is observed that the finetuned models are all highly susceptible to very repetitive outputs. Especially non-English languages, and the effect is worsened when using blueprints. For example (repetition in bold): *"Blueprint: 31%. What percentage of women with high education are looking for an infant scale? | Verbalisation (Swahili): Miongoni mwa wanawake wadogo, 31% pekee ya wanawake walio na kipimo* ***cha juu cha juu cha juu cha juu...****"*

Since only 200 tokens are generated at test time, excessive repetition in the blueprint sometimes results in the generation never reaching the actual verbalisation. This means the blueprint cannot be cut before comparing the output to the reference, which is highly undesirable as the blueprint itself is not meant to be evaluated against the reference.

Therefore, a repetition penalty is applied during inference (not during finetuning) to alleviate this. This is simply Huggingface's *repetition_penalty*[3] parameter which can be passed to *model.generate*. The repetition penalty is a form of penalised sampling, as formulated in Keskar et al. (2019) and simply penalises tokens which have already been generated.

$$p_i = \frac{\exp(x_i/(T \cdot I(i \in g)))}{\sum_j \exp(x_j/(T \cdot I(j \in g))} \qquad I(c) = \theta \text{ if c is True else 1} \tag{5.1}$$

In Equation 5.1, T is the temperature, θ is the strength of the penalty, g is a list of generated tokens and p_i is the probability distribution for the next token.

An analysis of varying repetition penalties is performed on the English-finetuned $\text{mT5}_{\text{Small}}$ with blueprints, as the model and dataset are small which allows for a faster analysis. Figure 5.1 shows that as the repetition penalty is increased, performance on all metrics increases until 1.4, after which the metrics begin to decline. A θ of 1.0 is equivalent to no penalty. However, by inspecting outputs it is observed that a penalty of 1.2 is enough to stop most of the highly problematic repetitive outputs. The penalty is somewhat a blunt instrument, so it is not over-used here. Although the scores are higher,

[3] https://huggingface.co/docs/transformers/internal/generation_utils

a penalty of 1.2 is cautiously chosen for the rest of the experiments as it mitigates the main problem, without altering the normal generations excessively. This agrees with the findings of Keskar et al. (2019), who also note that a θ of 1.2 in Equation 5.1 strikes a good balance between keeping repetition to a minimum but maintaining fluent and sensible outputs.

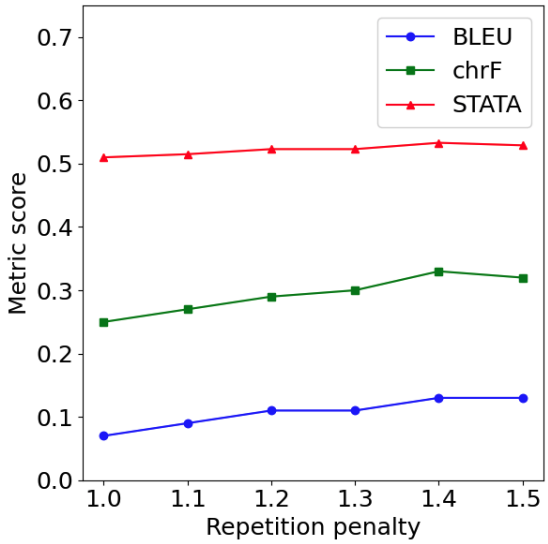


Figure 5.1: Multilingual model performance on the test set (including all languages) with varying repetition penalties.

Where highly repetitive blueprints do still occur at test time, these obviously bring down the metrics slightly, as a blueprint is being compared to a reference verbalisation which is not intended. But these bad candidates are not removed.

Chapter 6

Automatic Evaluation

Human evaluators would produce the best judgements on model outputs for TATA. However, this is infeasible for this project due to time, logistical and expense constraints. Evaluation is therefore done entirely through a combination of generic and learned automatic NLP metrics.

6.1 CHRF & BLEU

CHRF and BLEU are calculated and reported for the sake of interest, and to place in context how challenging this task is for both models and evaluation metrics. However, when interpreting them it must be kept in mind that they achieve extremely low correlations with human evaluations on TATA (<0.16) (Gehrmann et al., 2022).

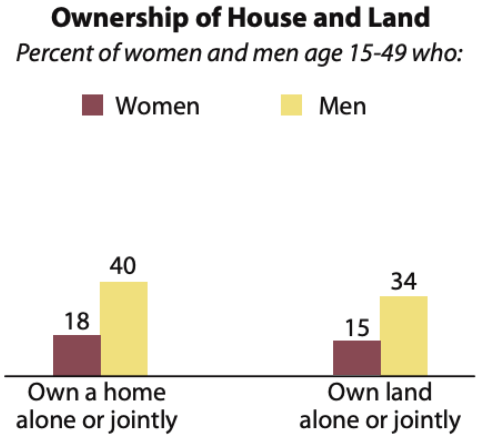


Figure 6.1: A chart from the DHS[1] which appears in TATA.

Why are popular automatic evaluation metrics so unsuitable for this task? Consider the following example. Figure 6.1 appears in TATA. Its reference verbalisations are:

- *"Only 18% of women own a house, either alone or jointly, and only 15% own land."*
- *"In comparison, men are more than twice as likely to own a home alone or jointly (40%)."*
- *"Men are also more than twice as likely to own land alone or jointly (34%)."*

Now, consider the rather simple candidate verbalisation, *"15% of women between the ages of 15 and 49 own land alone or jointly"*. This is a perfectly understandable

[1] https://dhsprogram.com/pubs/pdf/dm52/dm52.pdf

and attributable sentence in relation to the chart. In many ways it is a perfect output. However, its CHRF score, if we take the top score when comparing the candidate to all three references, is only 0.44.

This is partly due to the reasoning required, often over multiple cells, which simply cannot be captured by these metrics, and also due to the massive verbalisation space, i.e., the number of "correct" verbalisations is utterly vast compared to, for example, the number of correct translations of some sentence from one language to another. This is why metrics which consider the table itself, such as PARENT (Section 2.3) perform better on Table-to-Text.

Furthermore, consider the candidate verbalisation *"In comparison, men are more than twice as likely to own a home alone or jointly (20%)"*. This is syntactically nearly identical to the second reference, so it achieves a very high CHRF of 0.95. However, the number is incorrect (it is 20 but should be 40).

6.2 FACTKB

FACTKB differs fundamentally from metrics such as CHRF and BLEU as it was trained specifically to evaluate whether outputs are factually faithful to their sources.

FACTKB is intended to be used with *(summarisation, article)* input pairs. In this project, it is explored whether, and to what extent, it can be used to produce scores for *(verbalisation, linearised table)* pairs. FACTKB uses $\text{RoBERTa}_{\text{Base}}$ (Liu et al., 2019), which is not multilingual. Therefore it can only be used to evaluate the models finetuned on English examples.

To determine whether FACTKB can be used to evaluate Table-to-Text on TATA, scores are computed on examples from the human evaluations file, and the Pearson correlation between the score and the human judgement of attributability (0 or 1) is calculated. FACTKB achieves a Pearson correlation with human evaluations of **0.22**. This is still low, but better then the next-best performing existing metric, CHRF (0.16).

BERTScore, another metric designed for evaluating text generation, is also considered as it has been shown to be robust to challenging examples (Zhang et al., 2020). BERTScore calculates overlap between candidates and references like traditional metrics, but does so using contextual embedding similarity of tokens instead of exact token matches. It only achieves a Pearson correlation of **0.12** with human evaluations on TATA, which is lower than CHRF, so it is not used for evaluation.

6.3 STATA

Following instructions from Gehrmann et al. (2022), a learned metric is trained on human annotations of references and model outputs for TATA. It is called STATA, or *Statistical Assessment of Table-to-Text in African languages*.

The human annotations file marks outputs as "understandable" if they are fluent and grammatically correct, and "attributable" if they satisfy the "understandable" criteria,

and also only contain data which correctly reflects the data in at least one cell in the input table.

Gehrmann et al. (2022) define and train three STATA variants:

- **Ref**: A "traditional" metric model that uses just the candidate and the references.
- **QE**: A quality estimation model which considers the linearised input and the candidate, but not the references.
- **QE-Ref**: A quality estimation model which considers the linearised input, the candidate and the references. This setup is similar to the PARENT metric introduced in Section 2.3.

Gehrmann et al. (2022) showed that STATA performs significantly better than existing metrics for evaluating TATA. Gehrmann et al. (2022) designate STATA_{QE} as the dataset's main metric as it has the highest Pearson correlation, 0.66, with the human assessments. They state that using large mT5 models is necessary to achieve this high correlation. To illustrate this they also finetune mT5_{Base} for the same task, but it only achieves a correlation of **0.21** with human assessments. This is still better than the best-performing existing metric, CHRF, but not much. However, due to the same computational constraints mentioned in Section 5.1, $\text{mT5}_{\text{Large}}$ is used in this project instead. This yields a Pearson correlation with human assessments of attributability of **0.59**, significant at the $p < 0.01$ confidence level. While not as good as if the metric had been trained with mT5_{XXL}, this is still far better than any existing metric. Since STATA_{QE} was shown to perform the best, it is the only STATA variant replicated in the project.

It should be noted that the STATA scores reported in this project cannot be directly compared to those from Gehrmann et al. (2022), as they are trained with different mT5 checkpoints.

6.3.1 Training Details

$\text{mT5}_{\text{Large}}$ and $\text{mT5}_{\text{Small}}$ are finetuned for regression with an RMSE (Root Mean Squared Error) loss function. The metric is released on the Huggingface Hub (https://huggingface.co/adenhaus/mt5-large-stata) in the hope that it will prove useful to other researchers who wish to work on TATA.

A spare token in mT5's vocabulary is chosen as the regression token. At each step, the RMSE of this token's logit and the label, which is 0 or 1, is taken. This is the loss.

$$RMSE(y, \hat{y}) = \sqrt{\frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{N}} \tag{6.1}$$

In Equation 6.1, $\hat{y}$ is the logit of the special token and y is the label. N is the number of observations. At inference time, generation is constrained to this token, and its logit x is converted into a probability with the sigmoid, or logistic function (Equation 6.2). This is the final metric.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{6.2}$$

The human annotations file contains rows with the following fields and data types: output (string), model (string), interpretable (float), attributable (float), cells (float), reasoning (float), id (string), set (string), lang (string) and linearized_input (string).

Table 6.1 is an example row in Portuguese, with only the fields used for finetuning STATA.

output	attributable	linearized_input
Apesar de existirem pequenas diferenças no acesso a LLIN entre as áreas urbanas e as áreas rurais (54 por cento e 52 por cento, respetivamente), as variações no acesso a LLIN são significativas nas diferentes zonas de transmissão da malária.	1	Acesso a LLIN por parte da população do agregado familiar — Por cento — (Total) (53) (RESIDÊNCIA) (Urbana) (54) (Rural) (52) (ENDEMICIDADE DA MALÁRIA) (Epidémica nas terras altas) (62) (Endémica perto de lagos) (70) (Endémica costeira) (60) (Semi-árida, sazonal) (38) (Baixo risco) (40) (QUINTIL DE RIQUEZA) (Mais baixo) (37) (Segundo) (52) (Intermédio) (56) (Quarto) (55) (Mais elevado) (63)

Table 6.1: An example of relevant fields in a row of the human annotations data used to finetune STATA.

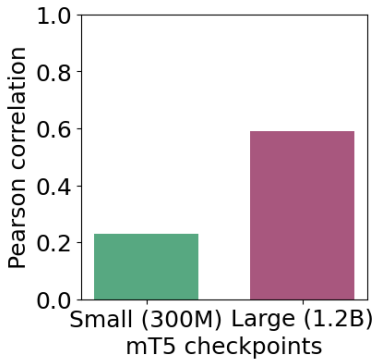


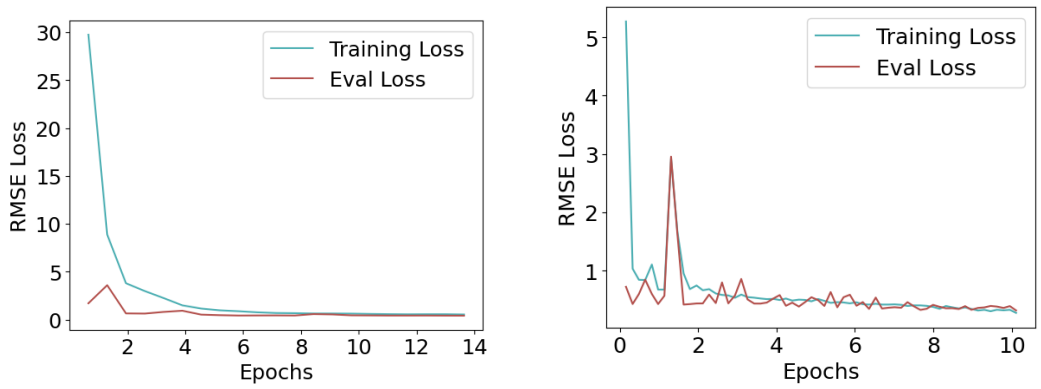
Figure 6.2: Pearson correlations of mT5-Small and mT5-Large STATA models with U+A human evaluations.

The file required some cleaning before it could be used for training. All rows where "attributable" is not either 0.0 or 1.0 are dropped. 73.7% of samples have a 0 in the "attributable" column, and 26.3% have a 1. This phenomenon, where the dataset is unbalanced in favour of the negative class, is common in classification or similar datasets. However, since the model learns the data quite effectively as-is, no data augmentation or other techniques are applied to balance the classes. The file is then randomly sampled into train (80%), validation (10%) and test (10%) splits. This makes the training set 4,900 rows and the validation and test sets 612 rows each. This dataset is also released on the Huggingface Hub (https://huggingface.co/datasets/adenhaus/stata) so that STATA can be easily retrained by other researchers, particularly those with more compute resources who can train it with mT5_{XXL}.

To evaluate the metric, scores are computed on the test set and the Pearson correlation between them and the human assessments of attributability (1, or 0) is computed. Figure 6.2 shows the correlations for the Small and Large model.

Hyperparameters & other details:

- Constant learning rate of 0.0001.
- Per-device batch size of 32 for Small and 8 for Large.
- 15 epochs.
- Dropout of 0.1.
- Inputs are truncated to 2048 tokens.
- Validation loss is measured every 0.1 epochs and after training the checkpoint with the lowest RMSE loss is selected.
- Training is done on a single NVIDIA A40 GPU.



(a) mT5-Small (b) mT5-large

Figure 6.3: Training curves for STATA models. Training is only shown up until lowest eval loss. Both models are trained for 15 epochs.

6.3.2 Examples

The first row in Table 6.2 shows a high-scoring example, i.e. an output which is very understandable and attributable to the source table. Relevant parts of the table and output are in bold and colour-coded to show attributability. The output is quite interpretable and refers to married women, which is the correct topic of the input table. The output also claims that "75% of married women participate in decision making about their own health", which is correctly attributable to the table. Therefore, it has a very high STATA score of 0.727.

The second row is neither understandable, nor attributable to the source table. It is not a well-formed sentence, and the quoted figure "86%" does not appear in the table at all, it is hallucinated. The only link it has at all to the source is the reference to malaria. Therefore, this output gets a very low STATA score of 0.49.

These examples have been hand-picked to illustrate what good and bad outputs look like, and what STATA scores they should receive. STATA, with a correlation with human assessments of 0.59, is not a perfect metric and does not get it right every time.

Linearised input	**Output**	**Score**
Women's **Participation in Decision-Making** — **Percent** of **married women** age 15-49 who usually make specific decisions by themselves or jointly with their husband — **(Own health care, 75)** (Major household purchases, 81) (Visits to family or friends, 88) (All 3 decisions, 65) (None of the 3 decisions, 7)	Among **married women**, **75%** of **married women** **participate in decision making** about **their health** care.	0.727
Items to support quality provision of **malaria** services — (Guidelines: treatment, 72) (Guidelines: diagnosis, 71) (Trained staff: diagnosis/treatment, 57) (Trained staff: IPT, 33) (Guidelines: IPT, 30) (Malaria microscopy, 18) (mRDT capacity, 81) (Any diagnostic capacity, 84)	Among facilities had an improved **malaria**, 86% of facilities had an improved **malaria**.	0.490

Table 6.2: Examples of input tables, model outputs, and corresponding STATA scores. Colours denote where relevant information from the input table is present in the output verbalisation.

Chapter 7

Results

7.1 English Subset Results

Model	CHRF	BLEU	FACTKB	STATA
$mT5_{Small}$	0.33	0.15	0.24	0.513
$mT5_{Small}$ (Blueprints)	0.30	0.11	0.28	0.523

Table 7.1: Results for $mT5_{Small}$ finetuned and evaluated on the English subset of TATA.

In Table 7.1, CHRF and BLEU are calculated using the predicted and reference verbalisations (traditional Reference-based metrics), and FACTKB and STATA are calculated using the predicted verbalisation and the linearised input table (Quality Estimation, or reference-less, metrics).

The results show that the blueprints do improve the attributability of outputs for the model finetuned on English rows. For the Small model, STATA increases from 0.513 to 0.523. This is significant because the range of values outputted by STATA is relatively narrow; from about 4.9 to 7.2 based on observation (in theory it could be outputting values between 0 and 1). This narrow range is common for learned metrics, and means that an increase of 0.1 is quite large. If the metric is trained with a larger model, the range of observed values will increase as the model becomes more confident. F_{act}KB also increases by 0.04 for the blueprint model.

mT5-Large is also finetuned on the English subset, however it overfits very quickly and does not achieve representative results, due to the dataset's tiny size (training curves are included in Appendix A.1). The full TATA dataset is small to begin with, and the English subset (about eight times smaller than the whole dataset, at 902 training examples) appears to simply be too small to finetune the Large model on. Therefore, these results are discarded and not reported.

7.1.1 Examples & common patterns

Reference	Linearised input	Setup	Blueprint	Verbalisation
17 percent of births in Kenya are unwanted, and 26 percent are mistimed (wanted later).	Planning Status of Births |Percent |(Wanted then, 0.57) (Unwanted, 0.17) (Wanted later, 0.26)	Vanilla		The proportion of births were wanted at the time of delivery.
		Blueprint	wanted births. What was the most common form of births?	The proportion of births were wanted at births, from a low of 7% in 1990 to a high of 11% in 2012.
The 2008-09 data indicate a rise since 2003 in medical antenatal care coverage.	Trends in Receipt of Antenatal Care from a Skilled Medical Provider , Kenya 2003-2008 |Percentage of women with live birth in the past 5 years |(2003, 88) (2008-09-01 00:00:00, 92)	Vanilla		Although the proportion of women who have received antenatal care from a skilled provider.
		Blueprint	88%. What was the rate of antenatal care from a skilled provider in 2003?	The proportion of women with antenatal care from a skilled provider in 2003.

Table 7.2: Output examples from vanilla and blueprint models, colour-coded to show where relevant information from the table has been used in blueprints and verbalisations.

In the first example in 7.2, neither the vanilla nor the blueprint verbalisations are very good in terms of understandability or attributability. Both refer to the correct concepts, but the former makes no reference to any actual data, and the second entirely hallucinates the data. The blueprint is semi-correct, as "wanted then" is the most common category, at 0.57. Unfortunately, this data is not correctly referenced in the verbalisation at all.

In the second example, a very good blueprint has been formulated. It captures important information from the table and does so correctly, identifying the right figure (88%) and year (2003). However, again the verbalisation fails to use this blueprint effectively, mentioning the correct year, but not the figure. This is still an improvement over the vanilla verbalisation, which references neither.

Some frequent idiosyncrasies emerge upon manually examining model outputs. Models will commonly produce phrases like *"increased from 15 percent to 15 percent"*. This is a common verbalisation pattern in the training data which the model learns, but it is clear that it has not learnt what *"increase"* means in this context, as the numbers are the same. Similar mistakes in characterising a comparison between two numbers

occur relatively often, even if both numbers do appear in the input table. For example, *"mortality rate is 19, compared to 19"* This is indicative of a lack of reasoning. Note that the repetition penalty of 1.2, which is applied, does not stop this.

The majority of verbalisations begin with *"The percentage of..."*, *"The proportion..."* or a few other common phrases. This is just a reflection of common sentence structures in the training data and is not an issue per se, but it does raise the concern that even a more powerful and capable model will probably not generate highly diverse verbalisations from this dataset.

Verbalisations are also sometimes not fully formed. For example, *"The proportion of children under age 5 who are wasted or too short for their age."* or *"Although the proportion of women who have received antenatal care from a skilled provider."* These would be valid as the first half of verbalisations, but are not completed and therefore do not make sense.

Generally, the title and unit parts of the input tables seem to appear the most consistently and accurately in the verbalisations, with the actual data points less so. This is consistent with an observation made by Gehrmann et al. (2022).

7.2 Multilingual Results

Model	CHRF	BLEU	STATA
$mT5_{Small}$	0.32	0.16	0.552
$mT5_{Small}$ (Eng blueprints)	0.29	0.09	0.525
$mT5_{Small}$ (Trans blueprints)	0.30	0.12	0.542
$mT5_{Large}$	0.33	0.13	0.552
$mT5_{Large}$ (Eng blueprints)	0.24	0.04	0.519
$mT5_{Large}$ (Trans blueprints)	0.27	0.11	0.544

Table 7.3: Results of finetuned multilingual models on the test set (all languages).

The Small baseline model trained by Gehrmann et al. (2022) achieves a CHRF of 0.33, which is very close to the 0.32 achieved by the baseline Small model in this project (Table 7.3), so the results are considered to be successfully replicated.

The English blueprint setup performs poorly (Table 7.3). Inspecting outputs of models trained on the English blueprints, which have to be able to generate outputs that contain multiple languages (English and the target language) reveals that they sometimes mix languages up. In the following example, colours denote different languages: *"Among Tanzania, one-third of Tanzanian women in the United States had all the three or more antenatal care visits, asilimia 29 ya wanawake in the United States had all the three or more antenatal care visits."*

"asilimia 29 ya wanawake" is a Swahili phrase meaning *"29 percent of women"* which has been incorrectly generated in the middle of an English sentence. (In this example, the United States is also hallucinated and a clause is repeated, but those are separate issues).

The translated blueprints fare decisively better than the English ones across all metrics, but are still slightly worse than no blueprints (Table 7.3).[1]

7.2.1 Model size

Gehrmann et al. (2022) saw a large performance jump on TATA from $mT5_{Small}$ to $mT5_{XXL}$ (13B). This raises the question of why in this project, Large (1.2B) doesn't improve over Small. One likely hypothesis is model-wise double descent, the phenomenon whereby performance degrades as the model size increases to a point, then begins to improve again as the model size is increased further (Nakkiran et al., 2019). Even the full TATA dataset is small, and the Large model converges quickly or overfits. Increasing the model size by a factor of 10 would still result in a fast convergence, but the model will likely achieve a much lower loss before this happens.

7.3 Per-Language Analysis

Lang	Small	Small Blueprints	Large	Large Blueprints
En	**0.19** / 0.33 / **0.551**	0.15 / 0.34 / 0.529	0.17 / **0.37** / 0.538	0.10 / 0.29 / 0.549
Sw	**0.21** / **0.39** / **0.589**	0.17 / 0.36 / 0.569	0.16 / 0.38 / 0.581	0.13 / 0.30 / 0.585
Yo	***0.03*** / 0.13 / 0.567	***0.03*** / ***0.14*** / 0.563	***0.03*** / ***0.14*** / **0.577**	0.02 / ***0.14*** / 0.561
Fr	**0.17** / 0.36 / 0.528	0.11 / 0.33 / 0.526	0.14 / **0.38** / 0.526	0.12 / 0.31 / **0.529**
Pt	**0.17** / ***0.39*** / 0.527	0.16 / 0.34 / 0.518	0.15 / ***0.39*** / 0.512	0.15 / 0.32 / **0.531**
Ha	**0.17** / ***0.33*** / 0.526	0.12 / ***0.33*** / 0.523	0.12 / ***0.33*** / **0.546**	0.12 / 0.29 / 0.515
Ar	**0.14** / 0.32 / **0.539**	0.11 / ***0.33*** / 0.523	0.12 / ***0.33*** / 0.533	0.12 / 0.31 / 0.519
Ig	**0.20** / **0.35** / 0.596	0.17 / 0.32 / 0.584	0.16 / 0.34 / **0.605**	0.15 / 0.27 / 0.558

Table 7.4: Language-specific performance of mT5 multilingual models (BLEU / CHRF / STATA). Bold figures represent the best result for each metric in each row. Italic figures are ties.

For more granular insights, the multilingual models are evaluated on each language in the test set individually. (The Blueprint columns in this table are translated blueprints as these were shown to perform better in Table 7.3). These per-language evaluation results are noisy (Table 7.4). Broadly, blueprints rarely improve any of the metrics, and performance between the Small and Large model is very close.

Some interesting observations: Yorùbá and Igbo are both low-resource, but widely-spoken in Western Africa, particularly Nigeria. Yet Igbo performs exceptionally well, achieving the highest STATA score of any language, while Yorùbá languishes at the bottom of the table by some margin, at least in terms of BLEU and CHRF. Yorùbá actually does perform relatively well on STATA. It has complex characters, with many accents (for example, *"Nómbà ti àwon..."*), so it is possible that some issue with

[1] A Russian zero-shot evaluation is not included in this project as when the finetuned models, which do not see Russian in the training data, are tested on Russian, they do not output Russian but a mixture of languages seen in the training data instead. This needs further examination.

capturing this is causing the very low BLEU scores. Gehrmann et al. (2022) did not get such poor Yorùbá scores.

The lower-resource African languages also tend to benefit the most from an increase in model size. Igbo, Hausa and Yorùbá all achieve higher STATA scores with the Large model, and are the only languages to do so. This suggests that scale is of particular importance for the low-resource languages.

7.4 Blueprint Analysis

Model	CHRF	BLEU
Small Trans Blueprints (Multilingual)	0.27	0.07
Small Blueprints (English)	0.23	0.05

Table 7.5: CHRF and BLEU between predicted and gold blueprints on dev set.

Table 7.5 shows how closely predicted blueprints match reference blueprints in the dev set (the candidates and references are split on *"Verbalisation:"* and only the blueprints compared). The dev set is used as no blueprints are generated for the test set, because at test time only the verbalisations are compared.

Clearly, these scores are very low. It should be noted that achieving a high CHRF or BLEU on the blueprints, or indeed the entire output, is not the explicit goal of training. If it was, these statistics would be calculated on the dev set during finetuning and the best model chosen based on which checkpoint optimised them. However, there are many valid blueprints for any given table, especially when generating short verbalisations from large tables, where there is lots of data to choose from. Arguably more important, is whether blueprints are related to the input table, and whether verbalisations are related to their blueprints.

Table 7.6 provides a measure of this. It shows how well the models are able to generate blueprints, and to what extent the verbalisations use these blueprints. To do this, CHRF and BLEU are calculated between the linearised input and the blueprint to quantify how much information from the table is present in the blueprint. CHRF and BLEU are also calculated between the blueprint and the verbalisation to quantify how closely the output relies on the blueprint for content selection.

Again, note that the goal is obviously not to have blueprints which are exactly the same as the input tables, nor verbalisations which are exactly the same as the blueprints. So, just maximising these metrics is not desirable here. The point is to interpret the models' scores with respect to the dataset scores, as an indicator the extent to which the models exhibit desirable attributes of the dataset.

The **English dataset** and **Multilingual dataset** rows in Table 7.6 represent scores calculated directly on the respective training datasets created in Chapter 4, and showcase the best-case scores. The **English model** ($mT5_{Small}$ finetuned on the English subset with blueprints) and **Multilingual model** ($mT5_{Small}$ finetuned on the full dataset with translated blueprints) rows represent scores calculated on the respective test sets,

	Linearised input → Blueprint		Blueprint → Verbalisation	
	CHRF	**BLEU**	**CHRF**	**BLEU**
English dataset	0.28	0.02	0.61	0.24
English model	0.24	0.02	0.39	0.20
Multilingual dataset	0.25	0.02	0.43	0.13
Multilingual model	0.23	0.01	0.36	0.16

Table 7.6: CHRF and BLEU between linearised inputs, blueprints and verbalisations in the training data and model outputs.

and the models' generated blueprints and verbalisations (the generation is split on *"Verbalisation:"* and the blueprint and verbalisation separated).

It is immediately clear that even the English model is not able to generate blueprints that are as closely related to the linearised table as the references (CHRF of 0.24 versus 0.28). Furthermore, the similarity between the blueprints and the outputs is very low (CHRF 0.39, BLEU 0.20) compared to the dataset (CHRF 0.61, BLEU 0.24). This indicates that not only is the model struggling to generate blueprints that are as good as gold, but it is also failing to remain as faithful to its blueprint (as showcased in the second example in 7.2). In other words, the model's verbalisations have significantly less in common with its blueprints than the dataset's.

The multilingual model actually does a better job of producing verbalisations which rely on the blueprints. There is a smaller percentage drop in **Blueprint → Verbalisation** CHRF than for English, and BLEU actually improves slightly. It is clear from these findings that the model's verbalisations do not draw from its blueprints enough, as exemplified in the examples in Table 7.2. One way of encouraging the model to rely more on its blueprints would be to use a form of constrained decoding, which could help focus the model on using words which it generates in its blueprint.[2] This is not explored in this project and is left to future work.

Still, there is a fundamental disadvantage in the multilingual setup before training even begins. Note how much lower the multilingual dataset's **Blueprint → Verbalisation** score (CHRF 0.43, BLEU 0.13) is than the English dataset's (CHRF 0.61, BLEU 0.24). This indicates that, as predicted in Section 4.3, inaccuracies in translating the English blueprints into the target languages have made it challenging to create high-quality multilingual blueprints.

So, why do the multilingual blueprint models achieve slightly higher STATA scores than the English-only blueprint models if the multilingual blueprints are lower quality? These multilingual models are still trained on around eight times more training data, as they see the full dataset, not just the English subset. This allows the multilingual models to learn the general task better, despite the language-specific challenges, and is also the reason the multilingual model achieves higher BLEU and CHRF between predicted and reference blueprints in Table 7.5.

[2] https://huggingface.co/blog/constrained-beam-search

Chapter 8

Conclusions

The evidence suggests QA blueprints are effective for slightly improving the attributability of Table-to-Text outputs in English. More work needs to be conducted to validate this, with both a larger English Table-to-Text dataset such as ToTTo, and a larger model, such as mT5_{XXL} or the newly-released Gemma-7b.[1]

In the multilingual setup, English blueprints degrade performance significantly and sometimes cause models to mix up multiple languages in their verbalisations.

Translated blueprints fare better, but still worse than no blueprints at all. This is due to inevitable inaccuracies in machine translating the blueprints from English to the target languages in the dataset for training. These imperfect translated blueprints mean that the multilingual models do not have high-quality gold examples to learn from.

As a result, the models struggle to generate blueprints which are as closely related to the input tables as the dataset. This problem, as measure by BLEU, is more severe for multilingual models than English.

For the English results, an increase in STATA scores is observed, while generic automated metrics such as BLEU and CHRF decrease. This further confirms that these metrics are not suitable for evaluating TATA due to their very low correlations with human evaluations. Although FACTKB performs slightly better, is not recommended to be used to evaluate TATA in future work, as its correlation with humans is still quite low, and this is not the task it was designed for. The learned metric STATA should be the definitive judge of model performance on TATA when human evaluators are not available. As part of this project, STATA is released online in the hope that it will make doing research on TATA more accessible. However, this STATA version is trained with $\text{mT5}_{\text{Large}}$. It should ideally be retrained with mT5_{XXL}, released and standardised, because if each researcher trains their own version, comparing scores across papers will be impossible.

It is also observed that increasing model size results in larger gains for the low-resource languages.

[1] https://blog.google/technology/developers/gemma-open-models/

Multilingual Table-to-Text generation remains a very challenging task for neural models. Based on the findings of this project, recommendations for future work are as follows: Trial using LLMs to generate more synthetic training data in several languages. If high-quality synthetic data can be generated, and the dataset size increased, this will boost model performance (although LLMs will probably only be able to generate good examples in the higher-resource languages). Additionally, constrained decoding should also be explored as a way to make verbalisations utilise the blueprints more. Finally, TATA can be turned into a more constrained task by having human annotators highlight the cells in each table that are used in its reference verbalisation, as the ToTTo dataset does. This greatly reduces the valid answer space, especially for larger tables.

Bibliography

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

Anton Benz and Katja Jasinskaja. Questions under discussion: From sentence to discourse. *Discourse Processes*, 54(3):177–186, 2017. doi: 10.1080/0163853X.2017.1316038. URL https://doi.org/10.1080/0163853X.2017.1316038.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense x retrieval: What retrieval granularity should we use?, 2023.

Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1053. URL https://aclanthology.org/D16-1053.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

P. Culicover and L. McNally. *The Limits of Syntax*. Syntax and Semantics. Brill,

2020. ISBN 9789004373167. URL https://books.google.co.za/books?id=JgD1DwAAQBAJ.

Kordula De Kuthy, Madeeswaran Kannan, Haemanth Santhi Ponnusamy, and Detmar Meurers. Towards automatically generating questions under discussion to link information and discourse structure. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5786–5798, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.509. URL https://aclanthology.org/2020.coling-main.509.

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

Shangbin Feng, Vidhisha Balachandran, Yuyang Bai, and Yulia Tsvetkov. Factkb: Generalizable factuality evaluation using language models enhanced with factual knowledge, 2023.

Sebastian Gehrmann, Sebastian Ruder, Vitaly Nikolaev, Jan A. Botha, Michael Chavinda, Ankur Parikh, and Clara Rivera. Tata: A multilingual table-to-text dataset for african languages, 2022.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

Mihir Kale and Abhinav Rastogi. Text-to-text pre-training for data-to-text tasks. In Brian Davis, Yvette Graham, John Kelleher, and Yaji Sripada, editors, *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland, December 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.inlg-1.14. URL https://aclanthology.org/2020.inlg-1.14.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858, 2019. URL http://arxiv.org/abs/1909.05858.

Ivana Kruijff-Korbayová and Mark Steedman. Discourse and information structure. *Journal of Logic, Language, and Information*, 12(3):249–259, 2003. ISSN 09258531, 15729583. URL http://www.jstor.org/stable/40180348.

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

Alexandre Magueresse, Vincent Carles, and Evan Heetderks. Low-resource languages: A review of past work and future challenges, 2020.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. Mqag: Multiple-choice question answering and generation for assessing information consistency in summarization, 2023.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *CoRR*, abs/1912.02292, 2019. URL http://arxiv.org/abs/1912.02292.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. Planning with learned entity prompts for abstractive summarization. *Transactions of the Association for Computational Linguistics*, 9:1475–1492, 2021. doi: 10.1162/tacl_a_00438. URL https://aclanthology.org/2021.tacl-1.88.

Shashi Narayan, Joshua Maynez, Reinald Kim Amplayo, Kuzman Ganchev, Annie Louis, Fantine Huot, Anders Sandholm, Dipanjan Das, and Mirella Lapata. Conditional generation with a question-answering blueprint, 2023.

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*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.89. URL https://aclanthology.org/2020.emnlp-main.89.

Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049.

Ratish Puduppully, Li Dong, and Mirella Lapata. Data-to-text generation with content selection and planning. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33016908. URL https://doi.org/10.1609/aaai.v33i01.33016908.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

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

Sebastian Ruder. A Review of the Neural History of Natural Language Processing. http://ruder.io/a-review-of-the-recent-history-of-nlp/, 2018.

Sebastian Ruder. The State of Multilingual AI. http://ruder.io/state-of-multilingual-ai/, 2022.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.704. URL https://aclanthology.org/2020.acl-main.704.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

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*, pages 2253–2263, Copenhagen, Denmark, September 2017a. Association for Computational Linguistics. doi: 10.18653/v1/D17-1239. URL https://aclanthology.org/D17-1239.

Sam Wiseman, Stuart Shieber, and Alexander Rush. Challenges in data-to-document generation. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark, September 2017b. Association for Computational Linguistics. doi: 10.18653/v1/D17-1239. URL https://aclanthology.org/D17-1239.

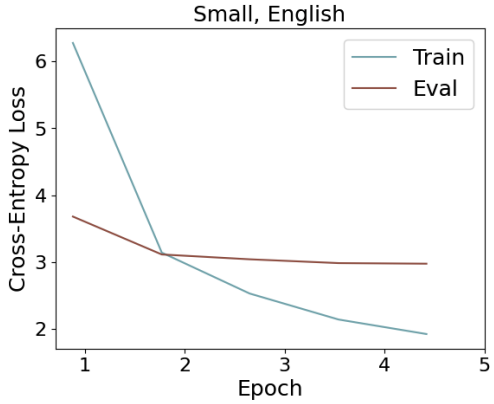
Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41.

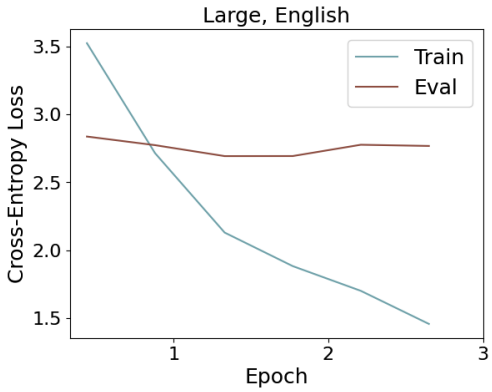
Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

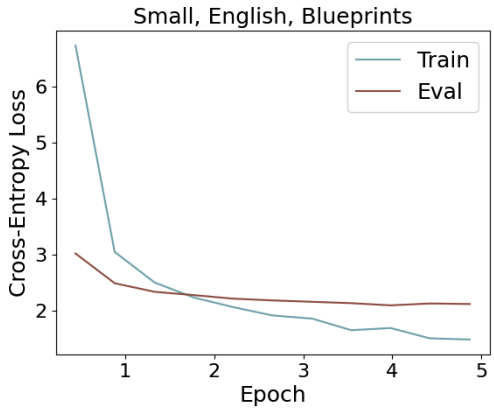
Appendix A

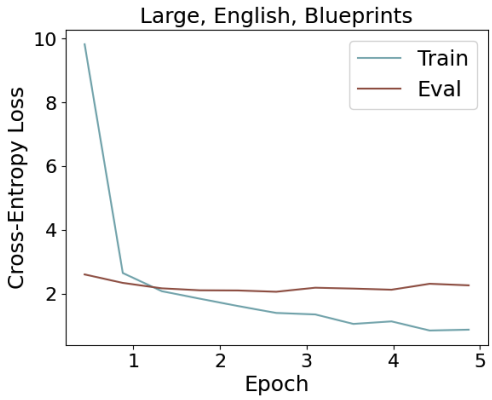
Training curves for all TaTA models

A.1 English Models

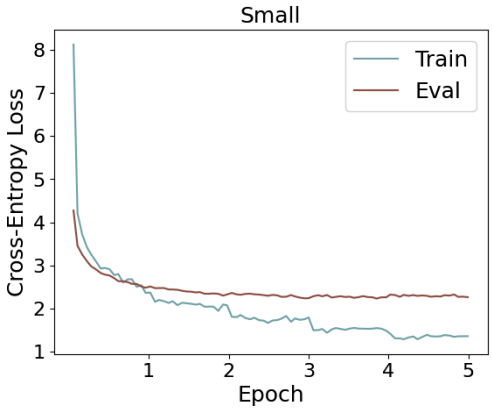


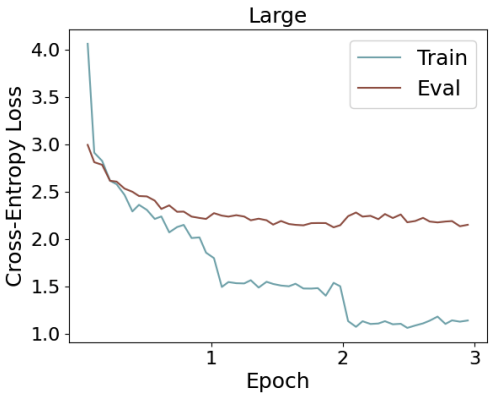


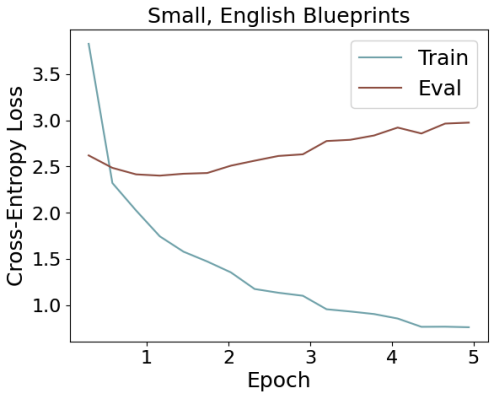


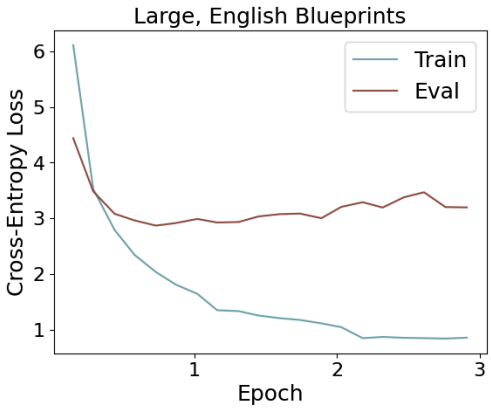


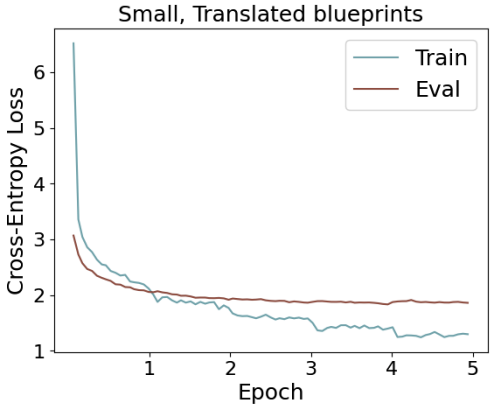
A.2 Multilingual Models

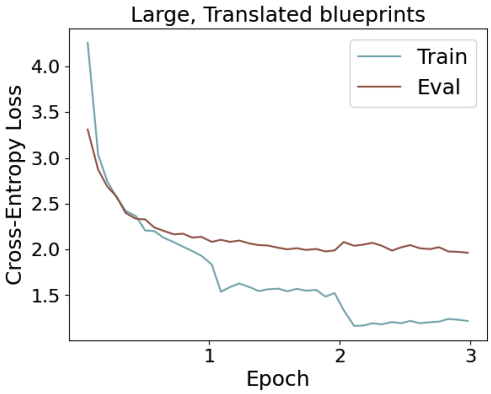

Small
Train
Eval
Cross-Entropy Loss
Epoch


Large
Train
Eval
Cross-Entropy Loss
Epoch


Small, English Blueprints
Train
Eval
Cross-Entropy Loss
Epoch


Large, English Blueprints
Train
Eval
Cross-Entropy Loss
Epoch


Small, Translated blueprints
Train
Eval
Cross-Entropy Loss
Epoch


Large, Translated blueprints
Train
Eval
Cross-Entropy Loss
Epoch