Triples-to-isiXhosa (T2X): Addressing the Challenges of Low-Resource Agglutinative Data-to-Text Generation

Francois Meyer and Jan Buys
Department of Computer Science
University of Cape Town
francois.meyer@uct.ac.za, jbuys@cs.uct.ac.za

Abstract

Most data-to-text datasets are for English, so the difficulties of modelling data-to-text for low-resource languages are largely unexplored. In this paper we tackle data-to-text for isiXhosa, which is low-resource and agglutinative. We introduce Triples-to-isiXhosa (T2X), a new dataset based on a subset of WebNLG, which presents a new linguistic context that shifts modelling demands to subword-driven techniques. We also develop an evaluation framework for T2X that measures how accurately generated text describes the data. This enables future users of T2X to go beyond surface-level metrics in evaluation. On the modelling side we explore two classes of methods — dedicated data-to-text models trained from scratch and pretrained language models (PLMs). We propose a new dedicated architecture aimed at agglutinative data-to-text, the Subword Segmental Pointer Generator (SSPG). It jointly learns to segment words and copy entities, and outperforms existing dedicated models for 2 agglutinative languages (isiXhosa and Finnish). We investigate pretrained solutions for T2X, which reveals that standard PLMs come up short. Fine-tuning machine translation models emerges as the best method overall. These findings underscore the distinct challenge presented by T2X: neither well-established data-to-text architectures nor customary pretrained methodologies prove optimal. We conclude with a qualitative analysis of generation errors and an ablation study.

Keywords: Less-Resourced/Endangered Languages, Natural Language Generation, Evaluation Methodologies

1. Introduction

Data-to-text is the task of transforming structured data (e.g. tables or triples) into text describing or summarising the data (Gatt and Krahmer, 2017). It is a valuable natural language generation (NLG) task, as it enables the separate evaluation of the text content (*what to say*) and its style (*how to say it*) (Wiseman et al., 2017). The majority of data-to-text datasets are in English or other high-resource languages, so such nuanced NLG evaluation is not possible for most low-resource languages.

Existing data-to-text models are designed for the linguistic typology of English. This is evident in that there are no studies on the role of subwords in data-to-text. Subwords are not essential for English data-to-text because English is morphologically simple — words are adequate units for modelling the complexity of datasets. Many examples in data-to-text datasets are instances of common templates. In English these are word-level templates (see Figure 1(a)). As a result, dedicated models for data-to-text do not apply subword segmentation, operating instead on word sequences (Wiseman et al., 2017, 2018; Shen et al., 2020).

This is not feasible for agglutinative languages like isiXhosa, where even simple templates are subword-based (see Figure 1(b)). The problem is compounded by the data scarcity of isiXhosa – heldout test sets have high proportions of new words, so subword modelling is essential. This paper studies data-to-text for isiXhosa: we create a data-to-text dataset with isiXhosa verbalisations, develop a data-focused evaluation framework, and investigate neural approaches for the task.

Data triple:
(South Africa, leaderName, Cyril Ramaphosa)
([subject], [relation], [object])

(a) English text and template:
Cyril Ramaphosa is the leader of South Africa
[object] is the [relation][subject]

(b) isiXhosa text and template:
uCyril Ramaphosa yinkokheli yoMzantsi Afrika
u[object] yin[relation] yo[subject]

Figure 1: Example from T2X, showing the need for subword-based data-to-text modelling.

IsiXhosa is one of South Africa's 12 official languages with over 8 million L1 speakers and 11 million L2 speakers (Eberhard et al., 2019). It is part of the Nguni languages, a group of related languages that are highly agglutinative and conjunctively written (morphemes are strung together to form long words). We present and release Triples-to-isiXhosa (T2X),[1] the first data-to-text dataset for any Southern African language. It was constructed by manually translating part of the English

[1] https://github.com/francois-meyer/t2x

LREC-COLING 2024, pages 16841–16854
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

WebNLG dataset and consists of triples of (subject, relation, object) mapped to descriptive sentences.

Alongside the release of T2X, we conduct a comprehensive investigation into neural data-to-text methods for low-resource agglutinative languages. We explore two prevailing directions of research: (1) LSTM-based encoder-decoder architectures designed to be trained from scratch for data-to-text (Wiseman et al., 2017, 2018; Shen et al., 2020), and (2) finetuning text-to-text pretrained language models (PLMs) (Kale and Rastogi, 2020; Nan et al., 2021; Ribeiro et al., 2021).

Data-to-text models trained from scratch are designed for word-based templates, which is inadequate for agglutinative languages like isiXhosa. We propose the subword segmental pointer generator (SSPG), a new neural model aimed at data-to-text for agglutinative languages.[2] It jointly learns subword segmentation, copying, and text generation. Our model adapts the subword segmental approach of Meyer and Buys (2022) for sequence-to-sequence modelling and combines it with a copy mechanism. SSPG learns subword segmentations that optimise data-to-text performance and copies entities directly where possible. We also propose unmixed decoding, a new decoding algorithm for generating text with SSPG.

We train SSPG on T2X and Finnish data-to-text (Kanerva et al., 2019a) (another agglutinative language). On both languages SSPG outperforms baselines trained from scratch: on T2X it improves chrF++ by 2.21 and BLEU by 1.11. These results show that *de facto* models for data-to-text are not well suited to the unique challenges posed by T2X. Our experiments on pretrained models yield similar conclusions. We finetune mT5 (Lewis et al., 2020) and Afri-mT5 (Adelani et al., 2022), but neither surpasses SSPG. We only see gains from pretraining when we turn to the unconventional strategy of finetuning English $\rightarrow$ isiXhosa translation models on T2X. So as in the case of models trained from scratch, well-established approaches to finetuning PLMs are suboptimal for T2X.

In addition to reporting automatic metrics, we develop an extractive evaluation framework for T2X that measures how accurately models describe data. Given output text, our framework estimates how well it describes triple data. This allows us to go beyond surface metrics like BLEU, evaluating the content of generations. We apply this framework to all our models, revealing tradeoffs between model capabilities. Subword segmental models copy entities more accurately, while standard subword models verbalise relations more effectively. Based on these findings, we qualitatively analyse the types of errors made by different models.

[2]Code and trained models available at https://github.com/francois-meyer/sspg.

2. Related Work

2.1. Neural Data-to-text

Traditionally data-to-text was framed as a series of subtasks (Reiter and Dale, 1997) handled separately through pipeline architectures (McKeown, 1992). This has been combined with deep learning (Puduppully et al., 2019; Puduppully and Lapata, 2021; Castro Ferreira et al., 2019). In our work we approach data-to-text as a sequence-to-sequence task for fully end-to-end learning. Such approaches can be categorised into neural architectures trained from scratch and finetuned PLMs.

Neural architectures Data-to-text is a highly structured NLG task, so there is room for exploiting this by equipping models with task-informed inductive biases. Wiseman et al. (2018) do this by inducing latent templates and generating text conditioned on these templates. Shen et al. (2020) model the segmentation of text into fragments aligned with data records. Both models are LSTM-based encoder-decoder models that use attention (Bahdanau et al., 2015) to incorporate a pointer generator into their decoder (Vinyals et al., 2015). This enables them to directly copy data tokens during text generation (See et al., 2017).

PLMs Finetuning text-to-text PLMs, such as BART (Lewis et al., 2020) and T5 (Raffel et al., 2020), has produced state-of-the-art results for data-to-text (Kale and Rastogi, 2020; Nan et al., 2021; Ribeiro et al., 2021). There are very few multilingual data-to-text datasets, so there has not been much work on finetuning multilingual PLMs like mBART (Liu et al., 2020) and mT5 (Xue et al., 2021). In the instances where this has been tried, such as for the Russian WebNLG dataset (Zhou and Lampouras, 2020) and the Czech Restaurant dataset (Dušek and Jurčíček, 2019), results have been promising (Gehrmann et al., 2021).

2.2. Subword Segmentation

Subword segmenters like BPE (Sennrich et al., 2016) and ULM (Kudo, 2018) operate separately from models trained on their subwords - they are applied in preprocessing. In the low-resource setting this leads to inconsistent performance (Zhu et al., 2019a) and oversegmentation (Wang et al., 2021; Ács, 2019). Similar issues arise for morphologically complex languages (Klein and Tsarfaty, 2020; Zhu et al., 2019b). These problems can be partially attributed to the separation of subword segmenters and model training. If segmenters produce suboptimal subwords, models cannot overcome this given insufficient training data.

Alternatively, segmentation can be cast as a latent variable marginalised over during training (Kong et al., 2016; Wang et al., 2017; Sun and

	Train	Valid	Test
WebNLG 1-triples	3 114	392	388
T2X triples	2 413	391	378
T2X verbalisations	3 859	600	888

Table 1: T2X dataset statistics

Deng, 2018; Kawakami et al., 2019). This leaves segmentation to the model - it is a learnable parameter for optimising the training objective. This has been used to learn subwords for MT (Kreutzer and Sokolov, 2018; He et al., 2020; Meyer and Buys, 2023) and low-resource language modelling (Downey et al., 2021; Meyer and Buys, 2022).

3. Triples-to-isiXhosa (T2X)

WebNLG (Gardent et al., 2017) consists of RDF triples from DBpedia paired with text verbalising the triples. Each example is one or more triples (up to seven) paired with a crowd-sourced verbalisation of one or more sentences. Multiple verbalisations are included for a large portion of the examples. The dataset has been expanded and translated into Russian, using machine translation and manual post-editing (Castro Ferreira et al., 2020). Recently translations have been released for Maltese, Irish, Breton, and Welsh.[3] These datasets cover smaller subsets of WebNLG with up to 1,665 examples per language (about half the size of T2X). Another data-to-text dataset, Table-to-Text in African languages (TATA) (Gehrmann et al., 2022), covers several languages including Swahili, which is a Niger-Congo B language like isiXhosa. TATA contains less than a thousand examples per language, and generation requires high-level reasoning about the data; in contrast our dataset primarily focuses on linguistic verbalisation ability. We are unaware of existing data-to-text datasets for Southern African languages.

Our dataset, Triples-to-isiXhosa (T2X), is based on the 1-triples in WebNLG version 2.1.[4] The choice to only include examples with single triples was motivated by the goal of obtaining a corpus covering a wide range of domains within the available annotation budget. The data covers 15 DBPedia categories. Three categories (Astronaut, Athlete and WrittenWork) are not included in the training data, only in the validation and test sets. Dataset statistics are given in Table 1 and example data-text pairs from the dataset are provided in Figure 1 and Tables 2 & 3. We publicly release the full dataset for use by future researchers.

[3]https://github.com/WebNLG/2023-Challenge
[4]https://huggingface.co/datasets/web_nlg

Annotation First language isiXhosa speakers who studied the language at university level were presented with triples and English WebNLG verbalisations, and asked to provide isiXhosa translations which reflect the content of the triples while phrasing the translations naturally. Annotators discussed questions arising during the process amongst each other, ensuring consistency among annotations. The verbalisations are relatively short, so translating them is an easy task given the isiXhosa proficiency of our annotators.

In the training and validation sets, only one isiXhosa verbalisation per triple is given for most domains, while the test set has multiple verbalisations (up to 3) for most examples. Multiple verbalisations correspond to different ways of describing the same data triple, capturing variations in phrasing for more nuanced evaluation. Equivalent verbalisations usually contain synonyms or different word orderings, as shown in the Table 2 example.

Data	(Germany, leaderName, Angela Merkel)
Text #1	Inkokeli yaseJamani ngu-Angela Merkel. *The leader of Germany is Angela Merkel.*
Text #2	U-Angela Merkel yinkokeli yaseJamani. *Angela Merkel is the leader of Germany.*

Table 2: An example T2X data triple mapped to two isiXhosa verbalisations (*with English translations*).

Task difficulty T2X maps single triples to isiXhosa verbalisations. Unlike WebNLG, it does not contain examples with multiple triples. In that sense it is a simpler task, not requiring combining information from multiple triples, but in some ways T2X is more challenging than existing datasets. For example, E2E (Novikova et al., 2017) covers one domain (restaurants). Much of the text follows a limited set of templates (e.g "[RESTAURANT NAME] is a [TYPE] restaurant in [AREA]"). T2X covers 15 domains and 286 relation types. In this sense T2X is quite challenging, since it would be difficult to model the dataset with a template-based approach. It requires some degree of generalisation and fluency, which is why learning-based approaches are more suitable.

T2X poses a different type of modelling challenge to English data-to-text datasets, because of the agglutinative nature of the isiXhosa language. In isiXhosa, morphemes are the primary units of meaning, so effective subword modelling is crucial. As shown in Figure 1, the underlying syntactic schemas for isiXhosa data-to-text generations are inherently subword-based. For English, a word-based model would cover most examples. For isiXhosa, a subword-based model is essential for even minimal text generation.

4. Subword Segmental Pointer Generator (SSPG)

In addition to benchmarking existing models, we propose SSPG to address the challenges of T2X. SSPG adds to the line of work designing models trained from scratch for data-to-text. While PLMs are widely used, dedicated data-to-text architectures remain valuable for low-resource languages with few available high-quality pretrained options. SSPG adapts the LSTM encoder-decoder — LSTMs are well suited to such low-resource tasks (Meyer and Buys, 2022) and persist as the preferred neural architecture for data-to-text (Wiseman et al., 2017, 2018; Shen et al., 2020). SSPG extends subword segmental modelling (Meyer and Buys, 2022), which was proposed as a modelling technique for agglutinative languages. SSPG simultaneously learns how to (1) map data triples to text, (2) segment text into subwords, and (3) when to copy directly from the data.

4.1. BPE-based Data Encoder

The encoder is a standard neural encoder for data-to-text: a bi-LSTM that processes data as flattened sequences of BPE tokens. BPE is applied to the data side of a data-to-text dataset. For example, the triple (France, currency, Euro) could be represented as the sequence "‹s _Fra nce s› ‹r currency r› ‹o _Euro o›". Special tokens delimit the boundaries between subject, relation, and object. BPE is sufficient for data-side segmentation, since most data-to-text datasets in other languages (T2X, Finnish Hockey, and translated WebNLG variants) have English data records.

4.2. Subword Segmental Decoder

The decoder is *subword segmental*, i.e., it jointly models the generation and subword segmentation of the output text. We follow the dynamic programming algorithm for subword segmental sequence-to-sequence training outlined by Meyer and Buys (2023), but modify their Transformer-based model to be LSTM-based and extend it to copy subwords.

During training SSPG processes data-text pairs $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ is a flattened triple of BPE tokens $\mathbf{x} = x_1, x_2, ..., x_{|\mathbf{x}|}$ and $\mathbf{y}$ is an unsegmented sequence of characters $\mathbf{y} = y_1, y_2, ..., y_{|\mathbf{y}|}$. We compute the probability of the output text conditioned on the input data as

$$p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{s}:\pi(\mathbf{s})=\mathbf{y}} p(\mathbf{s}|\mathbf{x}), \qquad (1)$$

where $\mathbf{s}$ is a sequence of subwords and π concatenates a sequence of subwords into its pre-segmented character sequence. Therefore we are marginalising over all possible subword segmentations of the output text $\mathbf{y}$.

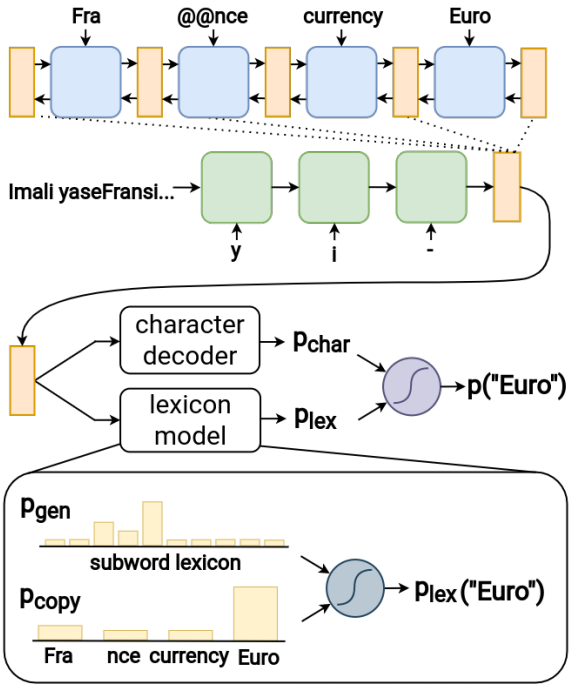

Figure 2: SSPG forward pass for (France, currency, Euro) → "Imali yaseFransi yi-Euro" ("The currency of France is the Euro"). At each character the next subword probability is computed with a mixture of a character-level decoder and a copy-equipped lexicon model (Eq. 2).

We model the probability of each subword sequence $\mathbf{s}$ with the chain rule, where each subword probability is computed with a mixture of a character-level decoder and a lexicon model as

$$p(s_i|\mathbf{y}_{<\mathbf{j}}, \mathbf{x}) = g_j p_{\text{char}}(s_i|\mathbf{y}_{<\mathbf{j}}, \mathbf{x}) + (1 - g_j) p_{\text{lex}}(s_i|\mathbf{y}_{<\mathbf{j}}, \mathbf{x}), \qquad (2)$$

where $\mathbf{y}_{<\mathbf{j}}$ is the character sequence up to the character immediately preceding the next subword s_i. The character model p_{char} is a character-level LSTM and the lexicon model p_{lex} is a softmax distribution over the subword lexicon, which consists of the top $|V|$ (a hyperparameter) most frequent character n-grams in training corpus. The coefficient g_j is computed by a fully connected layer. As shown in Figure 2, decoder probabilities are conditioned on the input data and the text history by passing attention-based decoder output representations to the subword mixture model components.

As shown in Figure 2, the output text history is encoded with a character-level LSTM to be tractable. This allows us to compute Eq. 2 at every character position in the output text. We extract probabilities for all subsequent subwords up to a specified maximum segment length and use dynamic programming to efficiently compute Eq. 1,

thereby summing over the probabilities of all possible subword segmentations of $\mathbf{y}$.

If we train this model to maximise Eq. 1, it optimises subword segmentation for its data-to-text task. This is the core idea of subword segmental modelling. It is valuable in settings, such as ours, where subwords are important enough to cast subword segmentation as a trainable parameter.

4.3. Copying Segments

The model outlined so far jointly learns data-to-text generation and subword segmentation. This could be useful for low-resource, agglutinative languages, and we include it as a baseline called subword segmental decoder (SSD). However, it is missing an essential component of most data-to-text models: the ability to copy entities from data, as achieved by pointer generator networks. We want to combine the strengths of subword segmental models and pointer generators.

We achieve this by including a conditional copy mechanism (Gulcehre et al., 2016) in the lexicon model p_{lex}. We introduce a binary latent variable z_j at each character j indicating whether the subsequent subword s_i is copied from data ($z_j = 1$) or generated from the lexicon ($z_j = 0$). We compute the p_{lex} in Eq. 2 by marginalising out z_j as

$$\begin{aligned} &p_{\text{lex}}(s_i|\mathbf{y}_{<\mathbf{j}}, \mathbf{x}) \\ &= p(z_j = 0|\mathbf{y}_{<\mathbf{j}}, \mathbf{x}) p_{\text{gen}}(s_i|\mathbf{y}_{<\mathbf{j}}, \mathbf{x}) + \\ &\quad p(z_j = 1|\mathbf{y}_{<\mathbf{j}}, \mathbf{x}) p_{\text{copy}}(s_i|\mathbf{y}_{<\mathbf{j}}, \mathbf{x}), \end{aligned} \tag{3}$$

where p_{gen} is a softmax layer over the lexicon and p_{copy} is the attention distribution over the data tokens. The probabilities $p(z_j = 0|\mathbf{y}_{<\mathbf{j}}, \mathbf{x})$ and $p(z_j = 1|\mathbf{y}_{<\mathbf{j}}, \mathbf{x})$ can be viewed as mixture model coefficients (similar to g_j in Eq. 2). They are computed by a fully connected sigmoid layer, allowing SSPG to learn (based on context) when it can rely on the lexicon's generation model and when it should look to the source to copy BPE tokens directly.

4.4. Unmixed Decoding

Standard neural models have one subword vocabulary, so beam search compares next-token probabilities from that vocabulary. However, subword segmental modelling uses a mixture model (Eq. 2), so it is not obvious how to use this for decoding. Meyer and Buys (2023) propose dynamic decoding, which combines information from the two mixture components (p_{char} and p_{lex} in Eq . 2). We initially used dynamic decoding to generate text with SSPG, but this resulted in weak performance. Our model's validation performance improved drastically when we developed a new decoding algorithm, which we call *unmixed decoding*.

We first describe the greedy version of the algorithm. SSPG subword probabilities are a mixture of three distributions: the character decoder, lexicon model, and copy mechanism. Unmixed decoding extracts next-subword probabilities from the three distributions *separately* and selects the next subword with the highest separated (*unmixed*) probability overall. At each decoding step we compute the top next-subword probability from each distribution as

$$\begin{aligned} p^*_{\text{char}} &= \max_s g p_{\text{char}}(s|\cdot), \\ p^*_{\text{gen}} &= \max_s (1-g) p(z=0|\cdot) p_{\text{gen}}(s|\cdot), \\ p^*_{\text{copy}} &= \max_s (1-g) p(z=1|\cdot) p_{\text{copy}}(s|\cdot), \end{aligned}$$

where we omit the conditioning variables of Equations 2 and 3 for simplification. We store the candidate subwords s^*_{char}, s^*_{gen}, s^*_{copy} corresponding to these probabilities. We then generate the subword corresponding to the highest probability among p^*_{char}, p^*_{gen}, p^*_{copy}. This process is repeated until the next subword is the end-of-sequence token. It is straightforward to combine unmixed decoding with beam search by extracting the top k subword candidates from each mixture component (resulting in $3k$ initial candidates), ranking these probabilities, and continuing with the top k subwords.

Each subword generated by unmixed decoding is put forward by one of the mixture components. During training SSPG learns in which contexts it should use the character decoder, generate from the lexicon, or copy a token from source. Unmixed decoding leverages this information during generation. For example, when the model is confident that the next subword should be copied from data, p^*_{copy} will be greater than p^*_{char} and p^*_{gen}, so the next subword is generated by the copy mechanism.

5. Experimental Setup

5.1. Models

We benchmark T2X with existing models and SSPG. Among our baselines, 3 are trained from scratch (PG, NT, SSD) and 5 are finetuned (mT5-base, mt5-large, Afri-mT5-large, BPE MT, SSMT). We tune hyperparameters for all models as detailed in Appendix A.

Pointer generator (PG) is an LSTM-based encoder-decoder model with a copy mechanism. This is commonly employed as a data-to-text baseline (Wiseman et al., 2017; Kanerva et al., 2019b).

Neural templates (NT) (Wiseman et al., 2018) learn latent templates for text. The LSTM-based decoder uses a hidden semi-markov model to jointly model templates and text generation.

Data	(a) (**South Africa**, capital, *Cape Town*)	(b) (**Christian Panucci**, club, *Inter Milan*)
Ref	Ikomkhulu lo**Mzantsi Afrika** li*Kapa*.	U**Christian Panucci** udlalela i-*Inter Milan*.
SSPG	I-*Cape Town* likomkhulu lase**South Africa**.	U**Christian Panucci** udlalela i-*Inter Milan*.
PG	U*Cape Town* likomkhulu lase-**Afrika**.	U**Christian Puucci** udlalela i-*Indter Milan*.
BPEMT	Ikomkhulu lo**Mzantsi Afrika** yi*Kapa*.	U**Christian Panucci** udlalela i-*Inter Milan*.
Data	(c) (**Ethiopia**, leaderName, *Mulatu Teshome*)	(d) (**Canada**, language, *English*)
Ref #1	U*Mulatu Teshome* yinkokheli yase-**Ethiopia**.	Isi*Ngesi* lulwimi oluthethwa e**Khanada**.
Ref #2	Igama lenkokheli e-**Ethiopia** ngu*Mulatu Teshome*.	Ulwimi lwesi*Ngesi* luthethwa e**Khanada**.
SSPG	U*Mulatu Teshome* yinkokeli yase-**Ethiopia**.	e**Canada** kuthetwa isi*Ngesi*.
PG	Inkokeli yase-**Ethiopia** ngu*Mulatu Teshome*.	Ulwimi lwesi*Ngesi* luthethwa e**Canada**.
BPEMT	U*Mulatu Teshome* yinkokeli yase-**Ethiopia**.	Isi*Ngesi* lulwimi oluthethwayo e**Canada**.

Table 3: Examples from T2X with model outputs. Subject verbalisations are **bold** and object verbalisations *italicised* to show that some entities should be copied directly while others should be translated. Green and red show correctly and incorrectly generated entities according to our evaluation framework.

Subword segmental decoder (SSD) is SSPG without a copy mechanism. We use dynamic decoding (Meyer and Buys, 2023) during generation.

mT5 (Xue et al., 2021) is the multilingual version of T5 (Raffel et al., 2020), covering 101 languages, including isiXhosa and Finnish.

Afri-mT5-base (Adelani et al., 2022) adapts mT5-base for 17 African language (including isiXhosa) through continued pretraining.

Bilingual pretrained MT (BPE MT & SSMT) Low-resource languages like isiXhosa are severely underrepresented in massively multilingual pretraining, so finetuning these PLMs does not guarantee good performance. Given the absence of existing NLG datasets, no research has investigated the effectiveness of pretraining and finetuning models for isiXhosa text generation. To further explore pretraining we turn to the only other publicly available pretrained encoder-decoder models for isiXhosa: machine translation (MT) models. We use bilingual MT models from Meyer and Buys (2023) for English $\rightarrow$ isiXhosa and English $\rightarrow$ Finnish, finetuning them on T2X and Finnish Hockey respectively. We finetune 2 MT models for each translation direction: a standard BPE-based model and SSMT (subword segmental machine translation), a model designed for agglutinative MT that jointly learns translation and subword segmentation.

5.2. Evaluation

Text overlap We compute several automatic metrics: BLEU (Papineni et al., 2002), chrF (Popović, 2015), chrF++ (Popović, 2017), NIST (Doddington, 2002), METEOR (Lavie and Agarwal, 2007), ROUGE (Lin, 2004), and CIDEr (Vedantam et al., 2015). Data-to-text presents an opportunity for more interpretable evaluation, such as quantifying how accurately generations reflect data content. To achieve this we adapt the extractive evaluation framework of Wiseman et al. (2017) for T2X.

Subject and object extraction In T2X (as in the other WebNLG translations) the triples are in English. Some entities can be directly copied from data (they are the same in English and isiXhosa), but others should be translated to isiXhosa. For example, in Figure 1 the name "Cyril Ramaphosa" should be copied, but the country "South Africa" should be translated to "Mzantsi Afrika".

To correctly verbalise data entities in T2X models have to learn when to copy entities and when to generate isiXhosa translations of entities. Some models *overcopy*, including English words in the output text where translations are required (see SSPG generation in Table 3(a)). Other models copy inaccurately, resulting in missing or partially copied entities in the text (see PG generation in Table 3(b)). We can quantify this by casting it as an information retrieval problem: does a generation contain the correctly verbalised entities?

For each example we check if the subject/object should be directly copied to the output text. We do this by checking if the entity string from the data triple is present in the reference text. If it is, the entity should be directly copied (like "Ethiopia" in Table 3(c)). If it is not, the entity should be translated (like "South Africa" in Table 3(a)). Each data-to-text example contains a binary decision for the subject and object, i.e., to translate or to copy. We test how well models learn this decision.

Relation prediction We cannot apply extraction to relations, because they cannot be copied from data. We follow Wiseman et al. (2017) in training a relation prediction model which we use to estimate how well models capture relations. We

	Model	chrF++	chrF	BLEU	NIST	MET	ROU	CID
Trained from scratch	PG	46.24	51.09	18.90	4.32	19.84	37.48	1.32
	NT	38.00	42.28	12.02	3.43	15.97	27.00	0.85
	SSD	44.77	49.91	16.16	4.14	19.85	34.97	1.16
	SSPG	**48.45**	**53.46**	**20.01**	**4.51**	**21.92**	**38.68**	**1.38**
Pretrained + finetuned	mT5-base	41.45	46.40	14.32	3.87	20.66	33.75	1.06
	Afri-mT5-base	42.42	47.64	16.11	4.02	20.49	34.10	1.15
	mT5-large	46.92	52.05	19.87	4.63	23.13	39.26	1.41
	BPE MT	**56.05**	**61.53**	**27.56**	**5.62**	**27.24**	**47.49**	**1.88**
	SSMT	54.01	59.44	24.19	5.33	26.13	44.34	1.61

Table 4: T2X test results. Best scores per category are **bold** and best scores overall are underlined.

		Subject			Object			Rel
	Model	P	R	F1	P	R	F1	acc
Trained from scratch	PG	71.30	63.37	67.10	**77.39**	74.40	75.86	**75.38**
	NT	72.65	73.25	72.95	67.14	68.12	67.63	38.14
	SSD	**76.01**	84.77	80.16	71.26	59.90	65.09	67.27
	SSPG	74.83	**88.07**	**80.91**	75.42	**85.99**	**80.36**	70.57
Pretrained + finetuned	mT5-base	70.27	85.60	77.18	73.79	73.43	73.61	53.45
	Afri-mT5-base	70.90	86.18	77.80	74.07	75.47	74.77	65.01
	mT5-large	70.78	**89.71**	79.13	75.22	82.13	78.52	69.37
	BPE MT	74.81	83.13	78.75	77.09	**84.54**	80.65	**87.69**
	SSMT	**77.78**	86.42	**81.87**	**83.17**	83.57	**83.37**	83.78

Table 5: T2X extractive results. Best scores per category are **bold** and best overall are underlined.

finetune *AfroXLMR-large* (Alabi et al., 2022) to predict relation types based on reference texts. The model achieves 85% heldout accuracy, which is high enough for estimating relation verbalisation capabilities. We apply this predictor to test set generations of models. We compare these to the correct relations in the test set data to estimate how accurately models describe relations.

5.3. Finnish Data-to-Text

To see if our findings generalise to another agglutinative language we perform experiments on Finnish data-to-text (to the best of our knowledge Finnish is the only other agglutinative language with a data-to-text dataset). The Finnish Hockey dataset (Kanerva et al., 2019a) is based on articles about ice hockey games. It contains game statistics and text spans that describe the corresponding game event. The data records are more complex than T2X (up to 12 records, depending on event type), but the texts are single sentences. Our models are trained on the 6,159 data records that are aligned with single text spans.

6. Results

6.1. Automatic metrics

The automatic metrics for T2X are reported in Table 4. SSPG outperforms the other dedicated data-to-text models trained from scratch across all metrics. We attribute the low scores of NT to the fact that BPE tokens are not ideal units for learning templates (NT is intended to learn word-level templates, but this led to even worse performance on T2X). Considering that PG outperforms SSD, learning subword segmentation is secondary to copying in terms of its importance. SSPG combines both aspects to achieve strong performance gains over PG, with increases of 2.21 on chrF++ and and 1.11 on BLUE. This establishes SSPG as a valuable dedicated neural architecture for data-to-text with agglutinative languages.

SSPG outperforms both mT5 variants on chrF++ and BLEU. We believe this is because mT5 is not pretrained on sufficient isiXhosa text and is intended primarily for less structured text-to-text tasks. Interestingly, SSPG even outperforms Afri-mT5. Adapted models like Afri-mT5 (e.g. Afro-XLMR-large (Alabi et al., 2022)) are considered strong pretrained baselines for African languages. The fact that it comes up short shows that stan-

	Model	chrF++	chrF	BLEU	NIST	MET	ROUGE	CID
Trained from scratch	PG	37.57	37.98	19.24	4.54	22.74	43.41	2.10
	NT	32.13	33.70	11.95	3.64	19.17	36.18	1.45
	SSD	33.68	33.78	17.03	4.17	21.53	41.48	1.88
	SSPG	**40.12**	**41.00**	**20.87**	**4.63**	**23.97**	**44.52**	**2.13**
Pretrained + finetuned	mT5-base	28.18	30.53	8.39	2.71	14.56	28.49	1.10
	mT5-large	32.70	34.22	13.40	3.46	19.48	39.93	1.75
	BPE MT	**42.37**	**41.59**	**22.36**	**5.04**	**24.87**	**46.04**	**2.25**
	SSMT	36.59	38.52	15.54	4.03	21.62	38.99	1.62

Table 6: Finnish test results. Best scores per category are **bold** and best scores overall are underlined.

(b)	Ref	UWilliam M.O. Dawson wazalelwa...
	PG	IWilliam M. O. Dawson wazalelwa...
(a)	Ref	UNorbert Lammert yinkokeli yaseJamani.
	PG	UNorbu Lammert yinkokeli yaseJamani.
(c)	Ref	I-Dublin yinxalenye yeLeinster.
	PG	IDubler yinxalenye yeLeinster.

Table 7: PG output compared to reference texts. Red shows where PG fails on subword copying.

Model and decoding algorithm	**chrF++**	**BLEU**
BPE +copy +beam search (PG)	48.25	18.81
Subword segmental models		
+copy +unmixed decoding (SSPG)	**49.41**	**20.35**
+copy +dynamic decoding	43.21	14.16
–copy +unmixed decoding	47.11	16.87
–copy +dynamic decoding (SSD)	46.84	17.54

Table 8: T2X validation scores for ablations.

dard pretrained solutions are not as reliable for low-resource text generation. We only see gains from pretraining whem we turn to the unconventional approach of finetuning MT models. The best model overall is the finetuned English → isiXhosa BPE MT model. This shows the value of pretraining + finetuning for this task, but highlights the lack of high-quality PLMs for isiXhosa. BPE MT outperforming SSMT shows that learning subword segmentation for this task is only advantagous when paired with a subword copy mechanism (our ablation results in Section 6.3 confirm this).

Table 6 shows the results for the Finnish Hockey dataset. The relative performance of models is similar to T2X. SSPG is the best dedicated model, while the finetuned English → Finnish BPE MT model is again best overall. Both models outperform the benchmark established by Kanerva et al. (2019a), with our best model (BPE MT) achieving a BLEU score gain of 2.69 over their PG model.

Based on our results, finetuning a pretrained MT model can be an effective approach to data-to-text modelling for certain languages. For high-resource languages finetuning text-to-text PLMs might yield better results, but for many low-resource languages MT models will be the best pretrained option available. For extremely low-resource languages with no available MT models, SSPG is the best choice for training a data-to-text model from scratch for agglutinative languages.

6.2. Extractive evaluation

Table 5 reveals a more mixed account of model performance based on the data content of model generations. Among the models trained from scratch, SSPG achieves the highest F1 scores. Its precision is lower than its recall, indicating some overcopying, but not to such an extent that it undermines its F1 scores. A comparison across all the individual precision and recall scores suggests that SSPG balances copying and translating better than the other models trained from scratch — it learns in which contexts to copy directly and when to translate instead.

PG outperforms SSPG on relations, which is based on descriptive isiXhosa text (e.g. "yinkokheli" in Fig. 1 means "is the leader"). The BPE subwords of PG are sufficient for modelling these isiXhosa phrases. However, PG struggles with the subword modelling of entities. isiXhosa has many prefixes that indicate grammatical roles (e.g. "*u*John" indicates singular personal proper noun). T2X requires attaching these prefixes to entities correctly. A qualitative analysis of generations reveals that PG struggles with this (see Table 7). SSPG does not seem to make these types of mistakes. By jointly modelling subword segmentation and copying, SSPG learns to combine the two when required.

As in the automatic metrics, SSPG outperforms mT5 but is outperformed by the MT models. SSMT achieves the highest F1 scores for subjects and objects, while BPE MT achieves the highest relation accuracy. This reiterates our findings for

the models trained from scratch: BPE subwords are sufficient for descriptive isiXhosa phrases, but modelling subword segmentaion allows SSMT to model subword-based changes to entities.

6.3. Ablation study

Table 8 shows the effect of different components on performance. A subword segmental encoder-decoder without a copy mechanism (SSD) falls well short of a BPE-based pointer generator (PG). Adding the copy mechanism improves performance but only if we use unmixed decoding, which is crucial for leveraging the copying ability of SSPG. While dynamic decoding is useful for high-resource tasks like MT, unmixed decoding is better suited for generating text with subword segmental models trained on smaller datasets.

7. Conclusion

We have presented Triples-to-isiXhosa (T2X), a new dataset for isiXhosa data-to-text. In addition we have proposed SSPG, a new neural model designed for agglutinative data-to-text. SSPG outperforms other dedicated data-to-text architectures on two agglutinative languages, isiXhosa and Finnish. SSPG is a strong data-to-text model for low-resource agglutinative languages without existing high-quality pretrained models. In the face of such resource scarcity we also explored fine-tuning bilingual NMT models, which produced the strongest results overall and should be further investigated as an alternative to PLMs. We publicly release T2X and our SSPG implementation to facilitate further research on isiXhosa text generation.

8. Ethics Statement

The source of the triples in our T2X dataset is the English WebNLG dataset. A large majority of the content covers Western people, places and events. The data content might also contain some biased, outdated or factually incorrect statements. This bias has a potentially negative impact on data-to-text models for isiXhosa developed based on the dataset, as some biases might be reflected the the model output and models might perform worse on non-Western people, places and events. Nevertheless, due to the limited availability of structured data in isiXhosa we belief that there is still a clear benefit to releasing this dataset to enable further development of data-to-text approaches for languages such as isiXhosa. Our model architecture, in particular through the copy mechanism, supports generalization to named entities different from those in T2X. Most of the relations covered by the triples (and verbalised in the isiXhosa text) are general enough. However, even with a copy mechanism our models sometime hallucinates and generate incorrect entities in the output, which limits the current reliability of the models.

9. Limitations

Our experiments are limited to two datasets for two languages. We cannot make claims about how well SSPG will generalise to other languages and differently structured data-to-text tasks. Our results are very similar across isiXhosa and Finnish and the differences in performance between models are substantial enough that we can confidently claim some generalisability, but only in a narrow linguistic context (simple data-to-text datasets for low-resource agglutinative languages). The limitations regarding the T2X dataset are discused in the ethics statement.

SSPG takes longer to train than PG because of the additional computation required by its dynamic programming algorithm for summing over latent subword segmentations. The increase in training time depends on the model's maximum segment length. Our final SSPG model has a maximum segment length of 5 characters and took aprroximately 4 hours to train on a single NVIDIA A100, as opposed to the 20 minutes training required for our final PG model.

Acknowledgements

We thank the annotators who assisted with creating the T2X dataset: Bonke Xakatha, Esethu Mahlumba, Happynes Raselabe, Silu Coki, Anovuyo Tshaka, Yanga Matiwane, and Phakamani Ntentema. We also thank Zukile Jama and Wanga Gambushe for discussions.

This work is based on research supported in part by the National Research Foundation of South Africa (Grant Number: 129850). Computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team: hpc.uct.ac.za. Francois Meyer is supported by the Hasso Plattner Institute for Digital Engineering, through the HPI Research School at the University of Cape Town.

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme,

Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Nikhil Baljekar, Xavier Garcia, Wolfgang Macherey, Theresa Breiner, Vera Saldinger Axelrod, Jason Riesa, Yuan Cao, Mia Chen, Klaus Macherey, Maxim Krikun, Pidong Wang, Alexander Gutkin, Apu Shah, Yanping Huang, Zhifeng Chen, Yonghui Wu, and Macduff Richard Hughes. 2022. Building machine translation systems for the next thousand languages. Technical report, Google Research.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 552–562, Hong Kong, China. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, page 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

C. M. Downey, Fei Xia, Gina-Anne Levow, and Shane Steinert-Threlkeld. 2021. A masked segmental language model for unsupervised natural language segmentation. *arXiv:2104.07829*.

Ondřej Dušek and Filip Jurčíček. 2019. Neural generation for Czech: Data and baselines. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 563–574, Tokyo, Japan. Association for Computational Linguistics.

David M. Eberhard, Gary F. Simons, , and Charles D. Fenning. 2019. *Ethnologue: Languages of the World*, 22 edition. SIL International.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2017. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad

Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

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

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149, Berlin, Germany. Association for Computational Linguistics.

Xuanli He, Gholamreza Haffari, and Mohammad Norouzi. 2020. Dynamic programming encoding for subword segmentation in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3042–3051, Online. Association for Computational Linguistics.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102, Dublin, Ireland. Association for Computational Linguistics.

Jenna Kanerva, Samuel Ronnqvist, Riina Kekki, Tapio Salakoski, and Filip Ginter. 2019a. Template-free data-to-text generation of finnish sports news. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa'19)*, Turku, Finland. Association for Computational Linguistics.

Jenna Kanerva, Samuel Rönnqvist, Riina Kekki, Tapio Salakoski, and Filip Ginter. 2019b. Template-free data-to-text generation of Finnish sports news. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 242–252, Turku, Finland. Linköping University Electronic Press.

Kazuya Kawakami, Chris Dyer, and Phil Blunsom. 2019. Learning to discover, ground and use words with segmental neural language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6441, Florence, Italy. Association for Computational Linguistics.

Stav Klein and Reut Tsarfaty. 2020. Getting the ##life out of living: How adequate are word-pieces for modelling complex morphology? In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 204–209, Online. Association for Computational Linguistics.

Lingpeng Kong, Chris Dyer, and Noah A. Smith. 2016. Segmental recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Julia Kreutzer and Artem Sokolov. 2018. Learning to segment inputs for NMT favors character-level processing. In *Proceedings of the 15th International Conference on Spoken Language Translation*, pages 166–172, Brussels. International Conference on Spoken Language Translation.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Kathleen McKeown. 1992. Text generation - using discourse strategies and focus constraints to generate natural language text. In *Studies in natural language processing*.

Francois Meyer and Jan Buys. 2022. Subword segmental language modelling for nguni languages. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6636–6649, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Francois Meyer and Jan Buys. 2023. Subword segmental machine translation: Unifying segmentation and target sentence generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2795–2809, Toronto, Canada. Association for Computational Linguistics.

Linyong Nan, Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Mutethia Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2021. DART: Open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 432–447, Online. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. 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. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. 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.

Ratish Puduppully and Mirella Lapata. 2021. Data-to-text Generation with Macro Planning. *Transactions of the Association for Computational Linguistics*, 9:510–527.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 211–227, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Xiaoyu Shen, Ernie Chang, Hui Su, Cheng Niu, and Dietrich Klakow. 2020. Neural data-to-text generation via jointly learning the segmentation and correspondence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7155–7165, Online. Association for Computational Linguistics.

Zhiqing Sun and Zhi-Hong Deng. 2018. Unsupervised neural word segmentation for Chinese via segmental language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4915–4920, Brussels, Belgium. Association for Computational Linguistics.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575. IEEE Computer Society.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Chong Wang, Yining Wang, Po-Sen Huang, Abdelrahman Mohamed, Dengyong Zhou, and Li Deng. 2017. Sequence modeling via segmentations. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 3674–3683. JMLR.org.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2021. Multi-view subword regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 473–482, Online. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187, Brussels, Belgium. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. 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. Association for Computational Linguistics.

Giulio Zhou and Gerasimos Lampouras. 2020. WebNLG challenge 2020: Language agnostic delexicalisation for multilingual RDF-to-text generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 186–191, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Yi Zhu, Benjamin Heinzerling, Ivan Vulić, Michael Strube, Roi Reichart, and Anna Korhonen. 2019a. On the importance of subword information for morphological tasks in truly low-resource languages. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 216–226, Hong Kong, China. Association for Computational Linguistics.

Yi Zhu, Ivan Vulić, and Anna Korhonen. 2019b. A systematic study of leveraging subword information for learning word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 912–932, Minneapolis, Minnesota. Association for Computational Linguistics.

Judit Ács. 2019. Exploring bert's vocabulary.

A. Hyperparameters

For each model we ran a grid search over varying hyperparameter settings. We chose the final model to evaluate on the test set based on validation chrF++ score, since it is well suited to evaluate morphologically rich languages (Popović, 2017; Bapna et al., 2022). Some of the hyperparameters are common to all models (learning rate, dropout, batch size, layers, embedding and hidden size). Other hyperparameters are unique to specific models. For NT we tuned the number of discrete states and the maximum length of segments. For SSD2T and SSPG we tuned the maximum segment length and the lexicon size. For BPE MT and SSMT we tuned the number of warmup updates and the learning rate scheduler.

We tried training our baselines without any subword segmentation, as reported in earlier papers

Model	lr	dropout	batch sz	BPE size	other
PG	1e-3	0.3	4	500	
NT	0.5	0.3	4	500	discrete states: 10, max segment length: 20
SSD2T	1e-3	0.3	4	250	lexicon size: 250, max segment length: 5
SSPG	1e-3	0.5	4	1k	lexicon size: 1k, max segment length: 5
mT5-base	1e-4	0.1	8	250k	warmup updates: 0, lr scheduler: fixed
mT5-large	1e-4	0.1	8	250k	warmup updates: 0, lr scheduler: fixed
BPE MT	1e-4	0.3	16	10k	warmup updates: 500, lr scheduler: inverse sqrt
SSMT	1e-4	0.3	16	5k	warmup updates: 0, lr scheduler: fixed

Table 9: Hyperparameter settings for our final T2X models, chosen based on validation chrF++ scores. Some hyperparameters are the same for all our models trained from scratch, including the number of LSTM layers in the encoder & decoder (1) and the size of the embedding & hidden layers (128).

for some of our baselines (NT and PG). However, for all our models we observed improved validation performance when BPE was used for subword segmentation before training. For NT and PG we trained BPE with a shared vocabulary on the data-to-text dataset. For SSD2T and SSPG we trained it on the data half of the data-to-text dataset, since we only use BPE to segment the data in these cases. We tuned BPE vocabulary size separately for each baseline over the range [250, 500, 1k, 2k, 5k] (see Table 9 for final vocabulary sizes). For the pretrained baselines we used their pretrained subword segmenters.