ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230512

XFLT: Exploring Techniques for Generating Cross Lingual Factually Grounded Long Text

Bhavyajeet Singh[*a], Aditya Hari[*a], Rahul Mehta[a], Tushar Abhishek[a,b], Manish Gupta[a,b], Vasudeva Varma[a]

[a]IIIT Hyderabad, India, [b]Microsoft

Abstract.
Multiple business scenarios require an automated generation of descriptive human-readable long text from structured input data, where the source is typically a high-resource language and the target is a low or medium resource language. We define the Cross-Lingual Fact to Long Text Generation (XFLT) as a novel natural language generation (NLG) task that involves generating descriptive and human-readable long text in a target language from structured input data (such as fact triples) in a source language. XFLT is challenging because of (a) hallucinatory nature of the state-of-the-art NLG models, (b) lack of good quality training data, and (c) lack of a suitable cross-lingual NLG metric. Unfortunately previous work focuses on different related problem settings (cross-lingual facts to *short text* or *monolingual* graph to text) and has made no efforts to handle hallucinations. In this paper, we contribute a novel dataset, XLAlign with over 64,000 paragraphs across 12 different languages, and English facts. We propose a novel solution to the XFLT task which addresses these challenges by training multilingual Transformer-based encoder-decoder models with coverage prompts and grounded decoding. Further, it improves on the XFLT quality by defining task-specific reward functions and training on them using reinforcement learning. On XLAlign, we compare this novel solution with several strong baselines using a new metric, cross-lingual PARENT. We also make our code and data publicly available[1].

1 Introduction

Fact-to-text (F2T) generation [33] is the task of transforming structured data (like fact triples) into natural language. F2T systems have been shown to be critical in many downstream applications like automated dialog systems [42], domain-specific chatbots [27], open domain question answering [6], authoring sports reports [4], financial reports [29], news reports [19], generating informative texts such as Wikipedia articles, etc. Unfortunately most of such systems are mono-lingual (typically English only) and also generate short text. Mono-lingual fact-to-text generation tends to suffer from the problem of data sparsity for low-resource languages.

Cross-lingual fact to *long* text (XFLT) systems could be useful across several business domains like healthcare, sports, travel, education, and reporting. In healthcare, English medical records can be used to generate patient summaries in regional languages. Drug information leaflets can be curated in different languages from English ingredients and effects. Summary of health insurance policies can be generated in different languages from English terms and conditions. English facts and warnings can be used to create public health alerts and advisories in different languages. Similarly, in sports, English statistics about events and players can be used to compose match reports, sports news, athlete biographies, and sports history essays in different languages. In tourism and travel, XFLT tools could help generate travel guides, hotel reviews, travel itinerary summary, travel blogs, travel advisories, travel-related news across languages given English facts.

Hence, in this paper, we study the XFLT task where the input is a bunch of English fact triples (subject, verb, object) related to an entity. The output is a paragraph in another target language which is expected to capture all the semantic information in English facts without hallucination. The solution is also expected to group related semantic information from facts into coherent sentences which appear in an appropriate order with smooth transitions. Fig. 1 shows an example.

The XFLT problem involves multiple challenges: (a) hallucination, (b) partially aligned training data, and (c) lack of an appropriate evaluation metric. NLG models have been notorious for generating hallucinatory text especially in long text generation settings. Further, although some labeled data exists for cross-lingual fact to *short* text (XF2T) [1], ground truth sentence is only partially aligned with input English facts. Only 10% of the sentences in the dataset have complete coverage with respect to their corresponding facts. Leveraging, such a dataset for cross-lingual fact to *long* text (XFLT) brings its own challenges. Lastly, while there exist source-dependent metrics like BLEURT [37] and PARENT [10], they are defined only for mono-lingual scenarios where input and output are in the same language. How do we define a similar source-dependent metric for our cross-lingual setting?

Like English fact-to-text (F2T) systems [18, 23, 41, 38, 26, 34, 46, 6, 13], recently there have been some efforts on multilingual and cross-lingual neural RDF verbalizers [1, 25, 14]. But they focus on generating short outputs, typically one sentence only. To the best of our knowledge, this is the first work that attempts to perform cross-lingual fact to *long* text generation. The ground truth generations in our dataset are 2.89 sentences long on average, where more than 40K examples have more than 3 sentences.

Given a dataset with partially aligned cross-lingual facts and sentences, our approach consists of two main modules: fact organizer and long text generator. Fact organizer clusters facts into logical groups and also predicts a sequence order over these groups. The long text generator is a multilingual Transformer-based encoder-decoder model with the following training recipe. The coverage prompts and grounded decoding tricks help us address the hallucination problem to a significant extent. Further, we obtain better quality

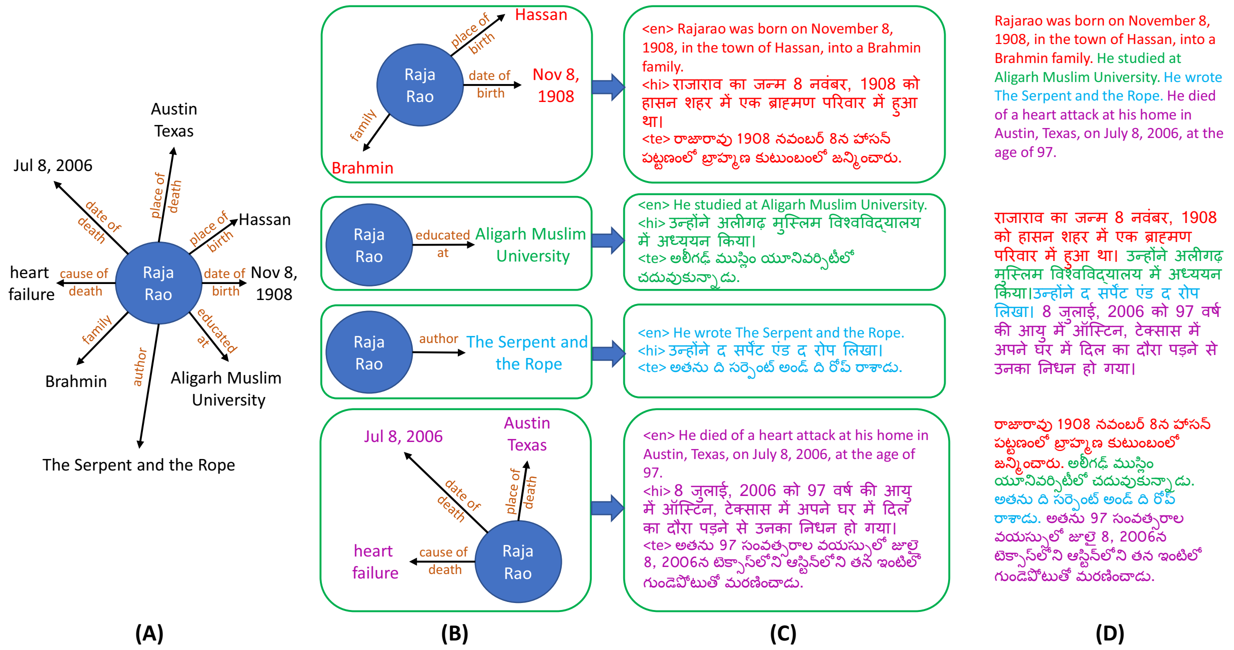

Figure 1. XFLT example: Generating English, Hindi and Telugu paragraphs to capture semantics from English facts

output with deep reinforcement learning (RL) using task-specific reward functions which motivate the model to generate outputs which are (a) syntactically aligned to ground truth output and (b) semantically aligned to input English facts.

Overall, in this paper we make the following contributions.

- We propose a novel problem: Cross-lingual fact to long text generation (XFLT), and a novel dataset, XLAlign.
- We propose a modular approach which uses coverage prompts and grounded decoding to reduce hallucination and deep reinforcement learning to improve quality.
- Our best model achieves a BLEU of 23 and cross-lingual PARENT score of 56. We make our code and data publicly available[1].

The remainder of the paper is organized as follows. We discuss related work in Section 2. We discuss details of the two modules of our proposed system in Section 3. We discuss dataset details, experiments and results in Section 4. Finally we conclude with a brief summary in Section 5.

2 Related Work

2.1 *Fact to Text Generation*

Initial F2T methods were template-based and were therefore proposed on domain-specific data like medical [2], cooking [9], person [11], etc. They align entities in RDF triples with entities mentioned in sentences, extract templates from the aligned sentences, and use templates to generate sentences given facts for new entities. Template-based methods are brittle and do not generalize well.

Recently, Seq-2-seq neural methods [18, 23] have become popular for F2T. These include vanilla LSTMs [41], LSTM encoder-decoder model with copy mechanism [38], LSTMs with hierarchical attentive encoder [26], pretrained Transformer based models [34] like BART [20] and T5 [32]. Richer encoding of the input triples has also been investigated using a combination of graph convolutional networks and Transformers [46], triple hierarchical attention networks [6], or Transformers with special fact-aware input embeddings [6]. Some recent work also explores specific F2T settings like plan generation when the order of occurrence of facts in text is available [46] or partially aligned F2T when the text covers more facts than those mentioned in the input [13]. Like our work, some studies [5, 31, 44] perform fact to long text generation. However, all of these methods focus on English F2T only.

2.2 *Cross-Lingual Fact to Short Text Generation (XFST)*

Our work is most related to fact verbalization tasks [1, 25, 14] where the focus is to use facts to generate short text. Abhishek et al. [1] perform cross-lingual fact to short text generation for 8 languages where each instance has 2.02 facts per instance and 19.8 words in the output text on average. Sagare et al. [35] extended this work to 12 languages. Gardent et al. [14] proposed the WebNLG dataset which contains data for English and Russian where each instance has 2.6 facts per instance and 23.7 words in the output text on average. Ferreira et al. [12] further enriched the corpus to include German as well. Moussallem et al. [25] verbalize RDF data to German, Russian, and English using the enriched WebNLG data, and experiment with an encoder-decoder architecture. As against these, we propose XFLT where the focus is on *long* text generation in a cross-lingual manner. Further, from a knowledge graph (KG) and text linking perspective, our work is related to tasks like entity linking (link mention in a sentence to a KG entity) [3] and fact linking (linking sentence to a set of facts) [16]. As against this, XFLT is the problem of generating a paragraph given a set of facts.

Recently there has been a lot of work on cross-lingual NLG tasks like machine translation [7, 22], question generation [8, 24], news ti-

[1] https://drive.google.com/file/d/1sHgcwXKribjrm2grbs-LzXUUqXQitD2N/

tle generation [21], and summarization [47, 39] thanks to models like XNLG [8], mBART [22], mT5 [45], etc. In this work, we investigate effectiveness of multiple modeling techniques for the XFLT task.

2.3 Source-Dependent Text Generation Metrics

Sai et al. [36] provide a survey of evaluation metrics used for NLG systems. Evaluation metrics for text generation like BLEU and ROUGE rely on the reference text. This is problematic when the reference and the source do not align entirely. Datasets for fact to text tasks are partially aligned, i.e., the reference text may have extra information not specifically mentioned in the input text. Hence, a source-dependent metric is suitable for fact to text tasks. Dhingra et al. [10] proposed PARENT as an NLG source-dependent metric that aligns n-grams from the reference and generated texts to the input text before computing their precision and recall. They show that PARENT correlates with human judgments better than other text generation metrics like BLEU, ROUGE, METEOR, CIDEr and CIDErD. However, PARENT works for monolingual tasks only since it relies on string matching. XFLT involves cross-lingual modeling and hence needs an adaptation of the PARENT metric for cross-lingual scenario. Hence, we propose XPARENT, which is a modified version of PARENT adapted for cross-lingual settings.

3 The Proposed Cross Lingual Fact to Long Text Generation System

Our dataset D containing N instances can be represented as $D = \{F_i, T_i, l_i\}_{i=1}^N$ where each instance D_i contains a set of $|F_i|$ English facts $F_i = \{f_j\}_{j=1}^{|F_i|}$ and an ordered list of aligned $|T_i|$ target sentences $T_i = [t_k]_{k=1}^{|T_i|}$ in the desired language l_i. A fact f_j is a tuple composed of subject s_j, relation r_j, object o_j and m qualifiers $Q = q_1, q_2, \ldots, q_m$. Each qualifier provides more information about the fact. Each of the qualifiers $\{q_j\}_{j=1}^m$ can be linked to the fact using a fact-level property which we call as qualifier relation qr_j. For example, consider the sentence: "Narendra Modi was the Chief Minister of Gujarat from 7 October 2001 to 22 May 2014, preceded by Keshubhai Patel and succeeded by Anandiben Patel." This can be represented by a fact where subject is "Narendra Modi", relation is "position held", object is "Chief Minister of Gujarat" and there are 4 qualifiers each with their qualifier relations as follows: (1) q_1="7 October 2001", qr_1="start time", (2) q_2="22 May 2014", qr_2="end time", (3) q_3="Keshubhai Patel", qr_3="replaces", and (4) q_4="Anandiben Patel", qr_4="replaced by". Further, the alignment between every target sentence t_k and set of English facts f_j is also provided as part of the dataset. We represent the aligned set of facts for target sentence t_k by $A(t_k)$.

Given the dataset D, our approach consists of a pipeline with two modules: fact organizer and long text generator. Fig. 2 shows the broad architecture of our proposed pipeline. We discuss details of these modules in this section.

3.1 Fact Organizer Training

For every instance $D_i \in D$, fact organizer clusters its facts $\{f_j\}_{j=1}^{|F_i|}$ into an ordered list of logical groups $G_i = g_1, g_2, \ldots, g_{|G_i|}$. Facts that align with a target sentence t_k, i.e., $A(t_k)$ should belong to the same logical group. Thus, ideally, there should be a logical group corresponding to each target sentence, i.e., $|G_i| = |T_i|$. Each logical group can consist of different number of facts. Also, each fact can belong to multiple logical groups.

We use an English Transformer-based encoder-decoder pretrained model for modeling the fact organizer. Each fact f_j is encoded as a string and the overall input consists of a concatenation of such strings across all facts in F_i. The string representation for a fact f_j is "$\langle S\rangle s_j \langle R\rangle r_j \langle O\rangle o_j \langle R\rangle qr_{j_1} \langle O\rangle q_{j_1} \langle R\rangle qr_{j_2} \langle O\rangle q_{j_2} \ldots \langle R\rangle qr_{j_m} \langle O\rangle q_{j_m}$" where $\langle S\rangle$, $\langle R\rangle$, $\langle O\rangle$ are special tokens. The overall input with F_i facts is obtained as follows: "cluster: f_1 f_2 $\ldots$ $f_{|F_i|}$". The overall output with G_i logical groups is obtained as follows: "$g_1 \langle BR\rangle g_2 \langle BR\rangle \ldots \langle BR\rangle g_{|G_i|}$" where each group g is a concatenation of constituent facts. Overall, the model is trained using the standard categorical cross-entropy loss L_{FO}.

The grouping of facts and the order in which these groups appear in the text is used as input for the long text generation.

3.2 Long Text Generator Training

The long text generator is a multilingual Transformer-based encoder-decoder model with the following training recipe. It uses coverage prompts to address the partially aligned nature of the training data. Further, it uses RL based training with reward functions to encourage grounded generations.

3.2.1 Coverage prompts to Reduce Hallucination

Ideally, in every instance of the dataset D, each target sentence t_k should contain the same semantic information as in its aligned set of facts $A(t_k)$. But practically, the set of aligned facts $A(t_k)$ may not cover the entire semantics of the target sentence t_k. We refer to this problem as partially aligned nature of the labeled data. If we train on such partially aligned data, the long text generator is encouraged to generate extraneous information beyond the semantics present in the input facts, leading to hallucination.

To address this problem, we first train a coverage classifier that estimates the degree to which the set of aligned facts $A(t_k)$ cover the semantics of the target sentence t_k. To train this classifier, we obtain coverage annotations for a part D_{cov} of the dataset D. Each target sentence t_k for every instance in D_{cov} is labeled with one of the two classes: complete coverage or partial coverage. The coverage classifier is a multilingual Transformer-based encoder with a classifier head which takes t_k and a string representation of $A(t_k)$ separated by a [SEP] token. Based on a threshold applied on confidence score with which the classifier predicts a fact-reference pair as completely aligned, we determine a coverage class (one of low, medium or high) for each of our training samples such that there are equal number of training instances for each of the classes per language.

While training the long text generator, we also incorporate the predicted coverage class as part of the input. Each training instance for long text generator model consists of a sentence t_k across all samples from D. At train time, we use the ground truth set of English facts aligned with t_k as input rather than using logical groups obtained from fact organizer. Overall, the input format for the long text generator is "generate l_i c_{ik}:" followed by a linearized string of facts in $A(t_k)$, where l_i is the target language of the sentence t_k and c_{ik} is the coverage class predicted using the coverage classifier. The long text generator is trained using the standard categorical cross-entropy loss L_{TG}. At inference time, we expect to generate sentences with high coverage and hence, we pass c always as "High" at inference time.

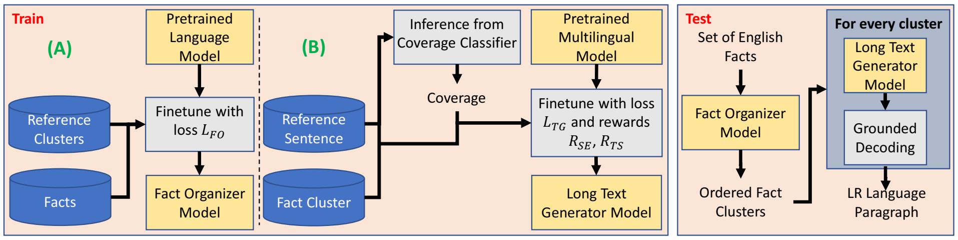

Figure 2. Proposed pipeline for cross-lingual fact to long text generation. Training involves finetuning (A) Fact Organizer Model and (B) Long Text Generation Model.

3.2.2 Reinforcement Learning for Improved Generation Quality

Further, we obtain better quality output with deep reinforcement learning using task-specific reward functions which motivate the model to generate outputs which are (a) syntactically aligned to ground truth output and (b) semantically aligned to input English facts.

Source Entailment Reward (R_{SE}): Given an instance with input as $A(t_k)$ and reference text t_k, source entailment reward measures the semantic similarity between the generated text and source English facts $A(t_k)$. The English fact tokens are not directly comparable with generated target language tokens. To bridge this gap, we introduce the notion of entailment probability, which is based on the probabilities that the presence of ngrams in the generated text is "correct" given the associated English facts. Estimating this probability is in itself a challenging language understanding task. Let y_k be the generated sentence text. Let y_k^n denote the list of all ngrams of y_k of order n. Let b denote one of such ngrams. Further, consider every token w in an ngram b. First, we compute entailment probability of token w being entailed by the source as the maximum of its probabilities of being entailed by each lexical item (subject, relation, object, or qualifier) v of a fact in the source.

$$P(w \Longleftarrow A(t_k)) = \max_{v \in A(t_K)} P(w \Longleftarrow v) \tag{1}$$

where $P(w \Longleftarrow v)$ is estimated by using similarity scores from MuRIL embeddings of the token w and lexical item v. Using this, we compute the entailment probability of ngram b being entailed as the geometric average of entailment probabilities of each of the constituent tokens as follows.

$$P(b \Longleftarrow A(t_k)) = \left(\prod_{w \in b} P(w \Longleftarrow A(t_k)) \right)^{1/|b|} \tag{2}$$

where $|b|$ is the order of the ngram b. Lastly, entailment score of generated sentence y_k for ngrams of order n with respect to the aligned ground truth facts is obtained by taking mean of entailment probabilities of each of the constituent ngrams as follows.

$$ES^n(y_k, A(t_k)) = \frac{\sum_{b \in y_k^n} (P(n \Longleftarrow A(t_k)))}{|y_k^n|} \tag{3}$$

where $|y_k^n|$ denotes the number of ngrams in y_k^n. Lastly, entailment score $ES(y_k, A(t_k))$ of generated sentence y_k with respect to the aligned ground truth facts is obtained by taking geometric mean of $ES^n(y_k, A(t_k))$ across all orders. The final source entailment reward is given by $R_{SE} = \lambda_{SE} \times ES(y_k, A(t_k))$ where λ_{SE} is a tunable hyperparameter controlling the importance of this reward in the overall objective to be optimized.

Target Similarity Reward (R_{TS}): This measures the syntactic similarity between the generated text y_k and reference text t_k. We measure this similarity using the BLEU metric. Thus, $R_{TS} = \lambda_{TS} \times BLEU(y_k, t_k)$ where λ_{TS} is a tunable hyperparameter controlling the importance of this reward in the overall objective to be optimized.

The rewards are used for policy learning. We employ the policy gradient algorithm [43] to maximize the expected reward (source entailment and/or target similarity) of the generated sequence y_k, whose gradient with respect to the parameters ϕ of the neural network model is estimated by sampling as follows.

$$\Delta_\phi J(\phi) = E[R.\Delta_\phi \log(P(y_k|x;\phi))] \tag{4}$$

where R is the R_{SE} reward and/or the R_{TS} reward, y_k is sampled from the distribution of model outputs at each decoding time step, x (which includes $A(t_k)$, language ID l_i and the coverage prompt) is the input to the model, and ϕ are the parameters of the long text generation model. The overall objectives for ϕ are the loss of the base model L_{TG} and the policy gradient of the different rewards.

3.3 Grounded Decoding during Inference

To reduce hallucination, at inference time, we use a decoding strategy that reduces the generation of text that is unsupported by the source, similar to [40]. This is based on the intuition that every word generated by the model should be entailed by the source facts, as long as the word captures some semantics from the source facts. Wrongly associating a content phrase (e.g. France) to the language model, simply because it seems more fluent (e.g. Paris France is fluent), might be a major cause of hallucination; since the facts may be discussing about the city of Paris in Texas, USA.

We encode this intuition in the decoding process as follows. At time t, while decoding the text y_k, we choose the top k tokens w based on their language modeling probabilities $P(w|y_{k[1:t-1]}, x; \phi)$. For each of these tokens w, we compute entailment probabilities $P(w \Longleftarrow A(t_k))$ using Eq. 1. Then, we perform beam search using a combination of these two probabilities as follows: $P(w|y_{k[1:t-1]}, x; \phi) \times P(w \Longleftarrow A(t_k))^{\lambda_{EF}}$ instead of just using the original language modeling probabilities.

3.4 Overall XFLT Inference

To summarize, the overall inference pipeline of our proposed system for XFLT works as follows. Given a set of English facts F_i for the i-th test instance, our fact organizer model outputs ordered fact clusters $G_i = g_1, g_2, \ldots, g_{|G_i|}$. Each fact cluster $\{g_k\}_{k=1}^{|G_i|}$ is then processed individually by our long text generator module along with grounded decoding to generate the output sentence y_k. Finally, these sentences are concatenated to generate the prediction paragraph $Y_i = concat(y_1, y_2, \ldots, y_k)$. Hyper-parameter details of various methods are provided along with the code.

4 Experiments and Results

4.1 Dataset

We derive our dataset, XLALIGN, from an existing dataset, XAlignV2 [35] (which is a revised version of XAlign [1]). XALIGNV2 is a cross-lingual fact to short text dataset with ∼0.55M (English facts, target language sentence) example pairs across 12 languages, of which 7425 pairs have been manually annotated. Example pairs corresponding to the same entity from XALIGNV2 are combined to obtain example (English facts, target language paragraph) pairs for our dataset, XLALIGN. The combination is done by a union of the English facts of corresponding XALIGNV2 examples, and a concatenation of sentences as per their order in the original Wikipedia article to create multi-sentence descriptions. In total, the XLALIGN dataset contains 125,106 paragraphs across 12 different languages. This is summarized in Table 1 which shows average number of facts, sentences, words per instance and instance counts in the train, validation, test splits. Compared to existing cross-lingual fact to short text datasets which contain one sentence per example, XLALIGN contains 2.9 sentences and 47.7 words on average.

Language	Instance Counts			Avg #Facts	Avg #Sents	Avg #Words
	Train	Val	Test			
Assamese (as)	799	159	111	7.0	4.3	66.9
Bengali (bn)	14,858	2,968	1,984	7.5	3.8	59.0
English (en)	32,176	6,427	4,292	5.3	2.4	41.2
Gujarati (gu)	901	179	121	6.0	3.3	55.6
Hindi (hi)	9,266	1,850	1,239	5.2	2.6	51.9
Kannada (kn)	2,026	404	273	6.6	3.7	51.1
Malayalam (ml)	8,363	1,671	1,117	6.0	3.2	40.4
Marathi (mr)	5,394	1,077	722	4.5	2.0	31.6
Odia (or)	1,742	348	237	6.9	4.1	63.0
Punjabi (pa)	5,454	1,085	731	6.5	3.1	84.1
Tamil (ta)	10,026	2,004	1,340	4.8	2.8	37.1
Telugu (te)	2,820	563	379	6.2	3.7	46.3
All	93,825	18,735	12,546	5.8	2.9	47.7

Table 1. Dataset statistics for the XLALIGN dataset.

XALIGNV2 contains examples with varying level of alignment between English facts and labeled target language sentences. This means that some semantics in the sentence is not captured by the corresponding facts. In order to quantify this partial alignment, we use scores from the coverage classifier described in Section 3.2.1 and illustrated in Fig. 3. This classifier was trained on binary labels obtained for 4376 examples. The classifier leads to a micro-averaged F1 of 0.9.

We split the dataset into train:validation:test in the ratio 75:15:10 as follows. To create a high-quality test and validation sets, the examples in XLALIGN were partitioned such that in the test and validation set, the ground truth target language paragraph contains least amount of extra information which is not covered by corresponding English facts. The train, validation and test split for each of the languages was also stratified based on the number of sentences per entity in the ground truth so that each of the splits contains equal proportion of paragraphs of different lengths.

After looking at the distribution of number of facts and sentences respectively across various languages in the XLALIGN dataset, we

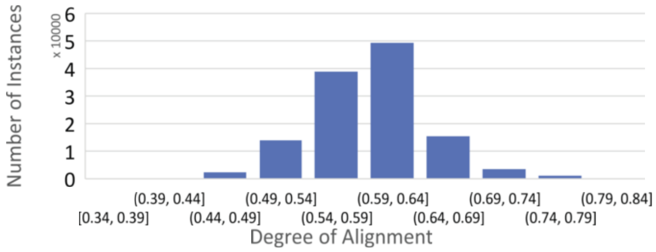

Figure 3. Distribution of degree of alignment across dataset instances in XLALIGN

Note that the dataset contains sizeable number of instances across various languages. Also, while creating the dataset we ensured that the number of sentences per example is limited to a maximum of 10 which leads to ∼1.6% examples with 20+ facts.

4.2 Metrics

We use two standard natural language generation metrics: BLEU [28][2] and chrF++ [30]. But these metrics rely on the reference text. This is problematic because in XFLT, the reference and the source do not align entirely, i.e., the reference text may have extra information not specifically mentioned in the input text. Hence, a source-dependent metric is suitable for XFLT. Further, since the task involves cross-lingual modeling, we propose XPARENT, which is a modified version of PARENT adapted for cross-lingual settings.

Given generated text y, target reference text t and corresponding source facts $A(t)$, we define XPARENT($y, t, A(t)$) as the F1 score (or harmonic mean) of entailed precision (EP) and entailed recall (ER) which in turn are defined as follows.

Entailed precision (EP) is computed as geometric average of entailed precision EP^n for ngrams of order n=1 to n=4. EP^n is further calculated as follows. Let y^n and t^n denote the list of all ngrams of order n of y and t respectively. Let b denote one of such ngrams in y^n. We consider the ngram b to be correct either if it occurs in the reference t, or if it has a high probability of being entailed by the source facts $A(t)$. Let $P(b \in t^n) = \min(\#(b, y^n), \#(b, t^n))/\#(b, y^n)$ where $\#(b, \circ)$ indicates number of times b occurs in $\circ$. Entailed precision EP^n for ngrams of order n is given by:

$$EP^n = \frac{\sum_{b \in y^n} [[P(b \in t^n) + P(b \notin t^n)P(b \Longleftarrow A(t))] \times \#(b, y^n)]}{\sum_{b \in y^n} \#(b, y^n)} \quad (5)$$

In words, an ngram receives a reward of 1 if it appears in the reference, with probability $P(b \in t^n)$, and otherwise it receives a reward of $P(b \Longleftarrow A(t))$ which is computed using Eq. 2. Both numerator and denominator are weighted by the count of the ngram in y^n. $P(b \in t^n)$ rewards an ngram for appearing as many times as it appears in the reference, not more.

Entailed recall (ER) is computed against both the reference ($ER(t)$), to ensure proper sentence structure in the generated text, and the input facts ($ER(A(t))$), to ensure that texts which mention more information from the facts get higher scores. These are combined using a geometric average as follows.

$$ER = ER(t)^{\lambda_R} ER(A(t))^{1-\lambda_R} \quad (6)$$

[2] Specifically, we use the implementation provided at https://github.com/mjpost/sacrebleu

	All Test Instances			Test Instances with $\geq$2 sentences		
	BLEU	chrF++	XPARENT	BLEU	chrF++	XPARENT
Single-Sentence XFST [1, 25]	15.515	45.410	42.202	14.059	44.171	40.301
Multi-Sentence XFST	18.660	37.621	50.338	15.873	37.067	50.327
Fact Organizer+Single-Sentence XFST	20.395	44.136	52.679	18.227	43.366	52.628
Fact Organizer+CP	22.060	48.821	55.271	18.442	48.119	55.074
Fact Organizer+CP+RL	22.663	49.532	55.328	18.760	48.717	54.966
Fact Organizer+CP+RL+GD	**23.010**	**50.142**	**56.555**	**19.036**	**49.318**	**56.132**

Table 2. Performance Comparison of various methods for XFLT task.

The parameter λ_R trades-off how much the generated text should match the reference, versus how much it should cover information from the facts.

Entailed recall $ER(t)$ with respect to reference t is computed as geometric average of $ER^n(t)$ for ngrams of order n=1 to n=4. We compute $ER^n(t)$ as follows.

$$ER(t) = \frac{\sum_{b \in t^n} [\min(\#(b, y^n), \#(b, t^n)) P(b \Longleftarrow A(t))]}{\sum_{b \in t^n} [\#(b, t^n) P(b \Longleftarrow A(t))]} \quad (7)$$

Entailed recall $ER(A(t))$ with respect to source facts $A(t)$ is computed at a word level as follows.

$$ER(A(t)) = \frac{\sum_{w \in A(t)} [I[P(w \Longleftarrow y) > \tau] \times \#(w, A(t))]}{\sum_{w \in A(t)} \#(w, A(t))} \quad (8)$$

where τ is a threshold tuned by manual inspection, w is a unique word in the concatenated string representation of facts in $A(t)$, $I[c]$ is the indicator function which takes a value of 1 if the condition c is true, else 0, and $P(w \Longleftarrow y)$ is computed using Eq. 1.

4.3 Fact Organizer Quality Evaluation

For our fact organizer, we use mT5-small. It provides a micro-F1 score of 0.595 and an MSE of 1.28 on average for prediction of the number of logical groups. For comparison, we also trained a MuRIL-base multi-class classifier to predict number of logical groups on XLAlign train set using categorical cross-entropy loss. This method provides much lower micro-F1 score of 0.245 and an MSE of 4.67 . Further, Fig. 4 shows the heatmap comparing actual versus predicted number of logical groups using the proposed fact organizer (left) and MuRIL-base classifier (right). From the heatmap as well as the micro-F1 and MSE values it is clear that a MuRIL-base classifier is poor at predicting the number of clusters.

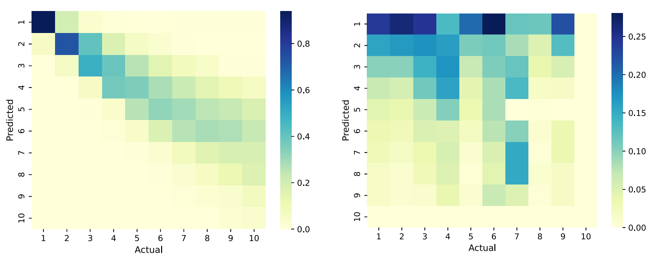

Figure 4. Heatmap comparing actual versus predicted number of logical groups using the proposed fact organizer(left) and MuRIL-base classifier(right).

Further, we wished to evaluate the quality of the discovered clusters using our fact organizer. We compute the quality as follows. First, given the discovered clusters and ground truth clusters, we compute 1:1 correspondence between them by modeling this as a linear sum assignment problem[3] and solve it using the Hungarian Method [17]. If number of discovered clusters is different from the number of ground truth clusters, the extra clusters on either side remain unassigned. Post the assignment, one can measure accuracy based on number of data points accurately clustered compared to ground truth. For our fact organizer, the average accuracy across test instances with $\geq$2 sentences turns out to be 81.49% which implies that our fact organizer is extremely effective at clustering facts into the expected logical groups.

Lastly, our fact organizer is also responsible for ordering the logical groups. To measure the quality of this ordering of logical groups, we can compare with the ground truth ordering of sentences. We perform this comparison using Kendall rank correlation coefficient (τ) [15] which is in the range [0,1] – higher the better. We find that the average Kendall-τ across test instances with $\geq$2 sentences turns out to be 0.696. This implies that our fact organizer not just discovers the right clusters but also sequences them in the expected order effectively.

4.4 Long Text Generator Quality Evaluation

For the long text generation, we use pretrained mT5-small as the base model architecture.

Baselines: Our work is closest to Cross-Lingual Fact to Short Text (XFST) methods. Hence, we compare our proposed method with two baseline approaches both of which also use the same base model architecture: Single-Sentence XFST and Multi-Sentence XFST. Multi-Sentence XFST is finetuned on XLAlign dataset where the input consists of a large number of English facts and the model is trained to generate multiple native language sentences. For training Single-sentence XFST model, we first split each instance in XLAlign train set such that each instance in the split dataset contains one native language sentence paired with the correspondence set of English facts. Single-Sentence XFST is then finetuned on this split dataset.

Ablations: Our full proposed method (Fact Organizer+CP+RL+GD) consists of several components: mT5 for clustering, coverage prompts, RL for improved generation quality and grounded decoding. To evaluate the importance of each component, we evaluate multiple ablations as follows: (1) Fact Organizer+Single-Sentence XFST: Coverage prompts, RL for improved generation quality and grounded decoding are removed. (2) Fact Organizer+CP: RL for improved generation quality and grounded decoding are removed. (3) Fact Organizer+CP+RL: Grounded decoding is removed.

Main Results: Table 2 shows performance comparison between the baselines, our proposed method and its ablations, on the XLAlign test set. We show BLEU, chrF++ and XPARENT for two settings: all test instances, and test instances with $\geq$ 2 sentences. While "all

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html

	Punjabi			English			Hindi			Marathi			Telugu		
	F	R	C	F	R	C	F	R	C	F	R	C	F	R	C
Ours	53	65	64	42	33	31	46	45	52	42	55	59	21	54	68
Multi-Sentence XFST	31	19	15	26	15	19	35	35	35	29	30	31	53	19	8
Both equal	16	16	22	32	52	50	19	21	13	29	15	10	26	27	24

Table 3. Human Evaluation: Percent times each method was preferred when compared to Multi-Sentence XFST baseline. F=Fidelity, R=recall, C=coherence.

Lang	Single-Sentence XFST						Fact Organizer+CP+RL+GD					
	All Test Instances			Test Instances with ≥2 sentences			All Test Instances			Test Instances with ≥2 sentences		
	BLEU	chrF++	XPARENT	BLEU	chrF++	XPARENT	BLEU	chrF++	XPARENT	BLEU	chrF++	XPARENT
as	5.092	34.406	26.786	5.035	34.062	26.613	8.119	43.359	40.311	7.232	43.538	41.362
bn	16.456	51.106	43.501	16.230	50.815	42.506	25.216	58.769	62.993	22.645	58.710	62.495
en	22.211	50.862	56.545	19.578	49.263	54.245	30.647	53.916	68.670	25.703	52.771	67.574
gu	6.621	32.977	29.204	6.109	32.454	28.235	13.598	40.644	43.824	10.578	39.945	45.501
hi	14.544	44.457	43.320	16.504	44.631	41.274	25.951	48.260	58.999	20.972	47.214	58.461
kn	4.280	31.220	21.893	4.200	30.769	21.428	7.551	36.216	39.051	6.426	36.141	40.650
ml	6.550	37.892	24.741	6.724	37.479	24.342	10.507	41.386	37.125	9.113	41.284	39.084
mr	22.529	41.051	40.656	12.057	33.124	32.993	29.859	51.130	56.449	18.502	45.948	51.947
or	17.632	52.457	42.941	18.114	52.218	42.990	26.598	60.014	50.528	26.848	60.352	52.334
pa	10.939	35.286	37.206	10.062	34.522	35.458	15.837	39.778	52.493	12.220	39.276	50.600
ta	6.637	42.681	22.951	5.850	41.774	21.592	11.912	44.941	36.687	9.124	45.140	37.933
te	3.863	29.620	24.246	4.118	29.391	23.887	8.488	39.591	38.409	7.112	39.465	40.101
All	15.515	45.410	42.202	14.059	44.171	40.301	23.010	50.142	56.555	19.036	49.318	56.132

Table 4. Language-wise Performance Comparison of the baseline XFST method and our proposed method.

test instances" contain ~33% instances with one sentence only (and is therefore similar to XFST setting), the "test instances with $\geq$ 2 sentences" is truly an XFLT setting.

We make the following observations from Table 2. (1) Results for the "test instances with $\geq$ 2 sentences" setting are typically lower compared to "all test instances" setting as expected. (2) Multi-sentence XFST is better than single-sentence XFST on BLEU and XPARENT. chrF++ is better for single-sentence XFST since its generations are relatively shorter and precise. (3) Fact Organizer helps improve the results for single-sentence XFST by a large margin. (4) Finetuning mT5 long text generator with coverage prompts leads to gains across all metrics. (5) RL based reward functions make the long text generator training more effective leading to gains across all metrics except XPARENT in the "test instances with $\geq$ 2 sentences" setting. We found that this minor decrease was because of a large decrease in entailed recall against the reference (ER(t)) for Tamil. We see consistent improvements across all metrics when using RL across all other languages. We also tried ablations using the two reward functions one by one, and found that both are needed for best results. (6) Finally, grounded decoding leads to the most accurate model. (7) All improvements for our full method (Fact Organizer+CP+RL+GD) are statistically significant compared to all baselines and ablations as measured using repeated measures ANOVA test with p-value < 0.05.

Language-wise Detailed Results for the Best Method: We show detailed language-wise results for the baseline XFST method and our proposed method (Fact Organizer+CP+RL+GD) on the XLAlign test set in Table 4. We observe that (1) Results with our proposed method (Fact Organizer+CP+RL+GD) are drastically better compared to the XFST method clearly showing that XFLT entails unique challenges different from XFST. (2) In the "All Test Instances" setting, BLEU improves relatively by 48.3%. On the other hand, in the "Test Instances with ≥2 sentences" setting, XPARENT sees the maximum relative improvement of 39.3%. (3) The biggest relative performance improvements are seen in Telugu, Gujarati and Kannada across metrics. Even in languages where XFST performed well, Fact Organizer+CP+RL+GD improves the metrics improves by >~1.5x.

4.5 *Qualitative Results*

Human evaluation results: For five languages, we compare Multi-Sentence XFST baseline with our best method. Evaluations were performed by 8 annotators (2 each for en, hi, te; 1 each for pa,mr). Each evaluator annotated 100 random samples for their respective native language. Table 3 shows the preference percentages based on fidelity, recall and coherence. Fidelity captures lack of hallucination. Recall captures how much of the semantics from facts were encoded in the output. Coherence assimilates how well the sentences are connected and how smooth is the flow of concepts in the generated output. We observe that in most cases, outputs from our proposed system are preferred over the best baseline.

Error Analysis: We manually examine 50 examples with low scores using our best method, to analyse the source of possible errors. We found that the most common source was the model repeating a set of words multiple times in a loop. Other sources included missing out facts from the input in the representation and generating extraneous information. Diverging references also lead to lower BLEU and chrF++ scores. Finally, we observed that the model has learned fact association patterns strongly. For example, even if the input facts do not have death cause but just have date of death, the model hallucinates the death cause. Since the model does not have any knowledge about the position of the sentence in the paragraph, in some cases, it generates pronouns in the first sentence and referent nouns in later sentences. This could be solved by passing in relative positional information as part of the model input in the future.

5 Conclusions

In this work we explored the XFLT problem for generation of multi-sentence paragraphs. We created a novel dataset, XLALIGN, using the existing XALIGNV2 dataset, with a high quality test partition. We explore different methods such as explicit clustering of facts, coverage prompting, grounded decoding and reinforcement learning each of which improve the quality of generation and address the problem of hallucination. These approaches can be used to directly generate Wikipedia like long text from structured data. We also define XPARENT score for evaluation of cross-lingual data-to-text problem which is of particular relevance for partially aligned ground truth text.

References

[1] Tushar Abhishek, Shivprasad Sagare, Bhavyajeet Singh, Anubhav Sharma, Manish Gupta, and Vasudeva Varma, 'Xalign: Cross-lingual fact-to-text alignment and generation for low-resource languages', in *The Web Conf*, pp. 171–175, (2022).

[2] K Bontcheva and Y Wilks, 'Automatic report generation from ontologies: the miakt approach', in *Conf. on application of natural language to info. systems*, pp. 324–335, (2004).

[3] J A Botha, Z Shan, and D Gillick, 'Entity linking in 100 languages', in *EMNLP*, pp. 7833–7845, (2020).

[4] D L Chen and R J Mooney, 'Learning to sportscast: a test of grounded language acquisition', in *ICML*, pp. 128–135, (2008).

[5] Mingda Chen, Sam Wiseman, and Kevin Gimpel, 'Wikitablet: A large-scale data-to-text dataset for generating wikipedia article sections', in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 193–209, (2021).

[6] W Chen, Y Su, X Yan, and W Y Wang, 'Kgpt: Knowledge-grounded pre-training for data-to-text generation', *arXiv:2010.02307*, (2020).

[7] Z Chi, L Dong, S Ma, S Huang, X-L Mao, H Huang, and F Wei. Mt6: Multilingual pretrained text-to-text transformer with translation pairs, 2021.

[8] Z Chi, L Dong, F Wei, W Wang, X-L Mao, and H Huang, 'Cross-lingual natural language generation via pre-training', in *AAAI*, volume 34, pp. 7570–7577, (2020).

[9] P Cimiano, J Lüker, D Nagel, and C Unger, 'Exploiting ontology lexica for generating natural language texts from rdf data', in *European Workshop on NLG*, pp. 10–19, (2013).

[10] Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen, 'Handling divergent reference texts when evaluating table-to-text generation', in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4884–4895, (2019).

[11] D Duma and E Klein, 'Generating natural language from linked data: Unsupervised template extraction', in *IWCS*, pp. 83–94, (2013).

[12] Thiago Castro Ferreira, Diego Moussallem, Emiel Krahmer, and Sander Wubben, 'Enriching the webnlg corpus', in *Proc. of the 11th International Conf. on Natural Language Generation*, pp. 171–176, (2018).

[13] Z Fu, B Shi, W Lam, L Bing, and Z Liu, 'Partially-aligned data-to-text generation with distant supervision', *arXiv:2010.01268*, (2020).

[14] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini, 'Creating training corpora for nlg micro-planning', in *55th annual meeting of the Association for Computational Linguistics (ACL)*, (2017).

[15] Maurice G Kendall, 'A new measure of rank correlation', *Biometrika*, **30**(1/2), 81–93, (1938).

[16] K Kolluru, M Rezk, P Verga, W W Cohen, and P Talukdar, 'Multilingual fact linking', in *AKBC*, (2021).

[17] Harold W Kuhn, 'The hungarian method for the assignment problem', *Naval research logistics quarterly*, **2**(1-2), 83–97, (1955).

[18] R Lebret, D Grangier, and M Auli, 'Neural text generation from structured data with application to the biography domain', in *EMNLP*, pp. 1203–1213, (2016).

[19] L Leppänen, M Munezero, M Granroth-Wilding, and H Toivonen, 'Data-driven news generation for automated journalism', in *INLG*, pp. 188–197, (2017).

[20] M Lewis, Y Liu, N Goyal, M Ghazvininejad, A Mohamed, O Levy, V Stoyanov, and L Zettlemoyer, 'Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension', in *ACL*, pp. 7871–7880, (2020).

[21] Y Liang, N Duan, Y Gong, N Wu, F Guo, W Qi, M Gong, L Shou, D Jiang, et al., 'Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation', *arXiv:2004.01401*, (2020).

[22] Y Liu, J Gu, N Goyal, X Li, S Edunov, M Ghazvininejad, M Lewis, and L Zettlemoyer, 'Multilingual denoising pre-training for neural machine translation', *TACL*, **8**, 726–742, (2020).

[23] H Mei, M Bansal, and M R Walter, 'What to talk about and how? selective gen. using lstms with coarse-to-fine alignment', in *NAACL-HLT*, pp. 720–730, (2016).

[24] Rajarshee Mitra, Rhea Jain, Aditya Srikanth Veerubhotla, and Manish Gupta, 'Zero-shot multi-lingual interrogative question generation for" people also ask" at bing', in *Proc. of the 27th ACM SIGKDD Conf. on Knowledge Discovery & Data Mining*, pp. 3414–3422, (2021).

[25] Diego Moussallem, Dwaraknath Gnaneshwar, Thiago Castro Ferreira, and Axel-Cyrille Ngonga Ngomo, 'Nabu–multilingual graph-based neural rdf verbalizer', in *The Semantic Web–ISWC 2020: 19th International Semantic Web Conf., Athens, Greece, November 2–6, 2020, Proceedings, Part I 19*, pp. 420–437. Springer, (2020).

[26] P Nema, S Shetty, P Jain, A Laha, K Sankaranarayanan, and M M Khapra, 'Generating descriptions from structured data using a bifocal attention mechanism and gated orthogonalization', in *NAACL-HLT*, pp. 1539–1550, (2018).

[27] J Novikova, O Dušek, and V Rieser, 'The e2e dataset: New challenges for end-to-end generation', *arXiv:1706.09254*, (2017).

[28] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 'Bleu: a method for automatic evaluation of machine translation', in *Proc. of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, (2002).

[29] V Plachouras, C Smiley, H Bretz, O Taylor, J L Leidner, D Song, and F Schilder, 'Interacting with financial data using natural language', in *SIGIR*, pp. 1121–1124, (2016).

[30] Maja Popović, 'chrf++: words helping character n-grams', in *Proc. of the second Conf. on machine translation*, pp. 612–618, (2017).

[31] Ratish Puduppully, Li Dong, and Mirella Lapata, 'Data-to-text generation with entity modeling', in *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2023–2035, (2019).

[32] C Raffel, N Shazeer, A Roberts, K Lee, S Narang, M Matena, Y Zhou, W Li, and P J Liu, 'Exploring the limits of transfer learning with a unified text-to-text transformer', *JMLR*, **21**, 1–67, (2020).

[33] E Reiter and R Dale, 'Building applied natural language generation systems', *NL Engineering*, **3**(1), 57–87, (1997).

[34] L F R Ribeiro, M Schmitt, H Schütze, and I Gurevych. Investigating pretrained language models for graph-to-text generation, 2021.

[35] Shivprasad Sagare, Tushar Abhishek, Bhavyajeet Singh, Anubhav Sharma, Manish Gupta, and Vasudeva Varma, 'Xf2t: Cross-lingual fact-to-text generation for low-resource languages', *arXiv preprint arXiv:2209.11252*, (2022).

[36] Ananya B Sai, Akash Kumar Mohankumar, and Mitesh M Khapra, 'A survey of evaluation metrics used for nlg systems', *ACM Computing Surveys (CSUR)*, **55**(2), 1–39, (2022).

[37] Thibault Sellam, Dipanjan Das, and Ankur Parikh, 'Bleurt: Learning robust metrics for text generation', in *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7881–7892, (2020).

[38] H Shahidi, M Li, and J Lin, 'Two birds, one stone: A simple, unified model for text generation from structured and unstructured data', in *ACL*, pp. 3864–3870, (2020).

[39] Dhaval Taunk, Shivprasad Sagare, Anupam Patil, Shivansh Subramanian, Manish Gupta, and Vasudeva Varma, 'Xwikigen: Cross-lingual summarization for encyclopedic text generation in low resource languages', in *Proc. of the ACM Web Conf. 2023*, pp. 1703–1713, (2023).

[40] Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh, 'Sticking to the facts: Confident decoding for faithful data-to-text generation', *arXiv preprint arXiv:1910.08684*, (2019).

[41] P Vougiouklis, H Elsahar, L-A Kaffee, C Gravier, F Laforest, J Hare, and E Simperl, 'Neural wikipedian: Generating textual summaries from knowledge base triples', *J. Web Semantics*, **52**, 1–15, (2018).

[42] T-H Wen, M Gasic, N Mrksic, L M Rojas-Barahona, P-H Su, D Vandyke, and S Young, 'Multi-domain neural network language generation for spoken dialogue systems', *arXiv:1603.01232*, (2016).

[43] Ronald J Williams, 'Simple statistical gradient-following algorithms for connectionist reinforcement learning', *Reinforcement learning*, 5–32, (1992).

[44] Sam Wiseman, Stuart M Shieber, and Alexander M Rush, 'Challenges in data-to-document generation', in *Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing*, pp. 2253–2263, (2017).

[45] L Xue, N Constant, A Roberts, M Kale, R Al-Rfou, A Siddhant, A Barua, and C Raffel, 'mt5: A massively multilingual pre-trained text-to-text transformer', in *NAACL-HLT*, pp. 483–498, (2021).

[46] C Zhao, M Walker, and S Chaturvedi, 'Bridging the structural gap between encoding and decoding for data-to-text generation', in *ACL*, pp. 2481–2491, (2020).

[47] J Zhu, Q Wang, Y Wang, Y Zhou, J Zhang, S Wang, and C Zong, 'Ncls: Neural cross-lingual summarization', in *EMNLP-IJCNLP*, pp. 3054–3064, (2019).