Generalising Multilingual Concept-to-Text NLG with Language Agnostic Delexicalisation

Giulio Zhou
Huawei Noah's Ark Lab
London, UK
giuliozhou@huawei.com

Gerasimos Lampouras
Huawei Noah's Ark Lab
London, UK
gerasimos.lampouras@huawei.com

Abstract

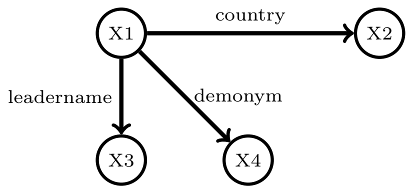
Concept-to-text Natural Language Generation is the task of expressing an input meaning representation in natural language. Previous approaches in this task have been able to generalise to rare or unseen instances by relying on a delexicalisation of the input. However, this often requires that the input appears verbatim in the output text. This poses challenges in multilingual settings, where the task expands to generate the output text in multiple languages given the same input. In this paper, we explore the application of multilingual models in concept-to-text and propose Language Agnostic Delexicalisation, a novel delexicalisation method that uses multilingual pretrained embeddings, and employs a character-level post-editing model to inflect words in their correct form during relexicalisation. Our experiments across five datasets and five languages show that multilingual models outperform monolingual models in concept-to-text and that our framework outperforms previous approaches, especially in low resource conditions.

1 Introduction

Recently, neural approaches to language generation have become predominant in various tasks such as concept-to-text Natural Language Generation (NLG), Summarisation, and Machine Translation thanks to their ability to achieve state-of-the-art performance through end-to-end training (Dušek et al., 2018; Chandrasekaran et al., 2019; Barrault et al., 2019). Specifically in Machine Translation, deep learning models have proven easy to adapt to multilingual output (Johnson et al., 2017) and have been demonstated to successfully transfer knowledge between languages, benefiting both the low and high resource languages (Dabre et al., 2020).

In the concept-to-text NLG task, the language generation model has to produce a text that is an accurate realisation of the abstract semantic information given in the input (Meaning Representation, MR; see Figure 1). It is common practice to perform a *delexicalisation* (Wen et al., 2015) of the MR, in order to facilitate the NLG model's generalisation to rare and unseen input; lack of generalisation is a main drawback of neural models (Goyal et al., 2016) but is particularly prominent in concept-to-text. Delexicalisation consists of a preprocessing and a postprocessing step. In prepro-

Input MR:



where: **X1**=amdavad ni gufa|амдавад ни гуфа, **X2**=india|индия, **X3**=t.s. thakur| т.с. тхакур (муж), **X4**=indian people|индийцы

Gold Target References:

amdavad ni gufa is located in **india**, where the leader is **t s thakur** and the demonym for people living there is **indian**.

амдавад ни гуфа находится в **индии** , где лидер **т.с. тхакур** и люди, проживающие там, называются **индийцами**.

Exact Delexicalisation:

X1 is located in **X2**, where the leader is t s thakur and the demonym for people living there is indian.

X1 находится в индии, где лидер т.с. тхакур и люди, проживающие там, называются индийцами.

Language Agnostic Delexicalisation (LAD):

X1 is located in **X2**, where the leader is **X3** and the demonym for people living there is **X4**.

X1 находится в **X2**, где лидер **X3** и люди, проживающие там, называются **X4**.

Figure 1: Delexicalisation on WebNLG Challenge 2020 with target output in English and Russian. Double underlining marks text missed by delexicalisation.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 114–127
August 1–6, 2021. ©2021 Association for Computational Linguistics

cessing, all occurrences of MR values in the text are replaced with placeholders. This way the model learns to generate text that is abstracted away from actual values. In postprocessing (relexicalisation), placeholders are re-filled with values.

The main shortcoming of delexicalisation is that its efficacy is bounded by the number of values that are correctly identified. In fact, a naive implementation of "exact" delexicalisation (see Figure 1) requires the values provided by the MR to appear verbatim in the text, which is often not the case. This shortcoming is more prominent when expanding concept-to-text to the multilingual setting, as MR values in the target language are often only partially provided. Additionally, MR values are usually in their base form, which makes it harder to find them verbatim in text of morphologically rich languages. Finally, relexicalisation also remains a naive process (see Figure 2) that ignores how context should effect the morphology of the MR value when it is added to the text (Goyal et al., 2016).

We propose Language Agnostic Delexicalisation (LAD), a novel delexicalisation method that aims to identify and delexicalise values in the text independently of the language. LAD expands over previous delexicalisation methods and maps input values to the most similar n-grams in the text, by focusing on semantic similarity, instead of lexical similarity, over a language independent embedding space. This is achieved by relying on pretrained multilingual embeddings, e.g. LASER (Artetxe and Schwenk, 2019). In addition, when relexicalising the placeholders, the values are processed with a character-level post editing model that modifies them to fit their context. Specifically in morphologically rich languages, this post editing results in the value exhibiting correct inflection for its context.

Our goal is to explore the application of multilingual models with a focus on their generalisation capability to rare or unseen inputs. In this paper, we (i) apply multilingual models and show that they outperform monolingual models in concept-to-text, especially in low resource conditions; (ii) propose LAD and show that it achieves state-of-the-art results, especially on unseen input; (iii) provide experimental analysis across 5 datasets and 5 languages over models with and without pre-training.

2 Related Work

Multilingual generation techniques have mostly been the focus of Machine Translation (MT) as the appropriate data (multilingual parallel source and target sentences) are more readily available there. Earlier research enabled multilingual generation with no and partial parameter sharing (Luong et al., 2016; Firat et al., 2016), while Johnson et al. (2017) explored many-to-many translation with full parameter sharing in a universal encoder-decoder framework. Despite the successes of this many-to-many framework, the improvements were mainly attributed to the model's multilingual input. Wang et al. (2018) improved on one-to-many translation (i.e. the input is always on a single language, while the output is on many) by introducing special label initialisation, language-dependent positional embeddings and a new parameter-sharing mechanism.

Generated Output before Relexicalisation:

X1 is located in **X2** where **X3** is the leader and the people are known as **X4**.

X1 находится в **X2**, где лидером является **X3**. местные жители известны как **X4**.

Exact Relexicalisation:

amdavad ni gufa is located in **india** where **t.s. thakur** is the leader and the people are known as **indian people**.

амдавад ни гуфа находится в **индия**, где лидером является **т.с. тхакур (муж)**. местные жители известны как **индийцы**.

Automatic Value Post-Editing (VAPE):

amdavad ni gufa is located in **india** where **t.s. thakur** is the leader and the people are known as **indians**.

амдавад ни гуфа находится в **индии**, где лидером является **т.с. тхакур**. местные жители известны как **индийцами**.

Figure 2: Relexicalisation examples; double underlining marks errors that ignore context.

In other language generation tasks, the vast majority of datasets are only available with English output. To enable output in a different language, a number of Zero-Shot methods have been proposed with the most common practice being to directly use an MT model to translate the output into the target language (Wan et al., 2010; Shen et al., 2018; Duan et al., 2019). The MT model can be fine-tuned on task-specific data when those are available (Miculicich et al., 2019). For the purposes of this paper, we do not consider these previous works as multilingual, as the language generation model is disjoint from the multilingual component, i.e. the pipelined MT model. Contrary to this, Chi et al. (2020) proposed a cross-lingual pretrained masked language model to generate in multiple languages, outperforming pipeline models on Question Generation and Abstractive Summarisation.

An adaptation of Puduppully et al. (2019) was applied to multilingual concept-to-text NLG and participated in the Document-Level Generation and Translation shared task (Hayashi et al., 2019, DGT). However, this shared task, and in extension the dataset and participating systems, heavily focus on content selection and document generation. Additionally, the input's attributes are constant across train and testing, so there are no unseen data and no need to improve on the model's generalisation capability. As the goal of this paper is multilinguality (content selection is a language agnostic task) and generalisation, we opt to not use this dataset.

Multilinguality has also been explored in the related tasks of Morphological Inflection and Surface Realisation in SIGMORPHON (McCarthy et al., 2019) and MSR (Mille et al., 2020) challenges. However, our Automatic Value Post-Editing approach focuses mostly on adapting values to context and does not assume additional input such as dependency trees, PoS tags or morphological information that Surface Realisation and Morphological Inflection often requires.

Particularly for concept-to-text NLG, notable previous works includes the approach of Fan and Gardent (2020) who make use of pretrained language models through the Transformer architecture for AMR-to-text generation in multiple languages, and the WebNLG Challenge 2020 (Castro Ferreira et al., 2020). The goal of WebNLG 2020 was to generate output in both English and Russian but most of the participants focused on monolingual rather than multilingual approaches.

3 Rare and Unseen Inputs in NLG

Due to the existence of open-set and numerical attributes in the aforementioned datasets, it is common during testing for MRs to contain rare or unseen values. Certain datasets are even more challenging in this regard (e.g. WebNLG Challenge 2020) as they also contain unseen relations in the development and test subsets. Several techniques have been proposed to mitigate this problem.

Delexicalisation, also known as anonymisation or masking, is a pre/post-processing procedure that attempts to mitigate problems with data sparsity. In preprocessing, all values in the MR that appear verbatim in the target sentence are replaced in both input and output with specific placeholders, e.g. "X-" followed by the corresponding attribute (e.g. "X-type") so that the placeholder still captures relevant semantic information. In Figure 1 we use numbered placeholders instead, for clarity and space. The model is trained to generate the target text containing these placeholders, which are subsequently replaced with the corresponding true values (i.e. relexicalised) in post-processing. See Figures 1 and Figure 2 for examples; we mark this strategy as *Exact* due to the exact matching of the values with the text. To improve delexicalisation accuracy, n-gram matching (Trisedya et al., 2018) has been proposed as an alternative. Thanks to its simplicity and efficacy, delexicalisation is widely used by many systems, including the winning systems of major concept-to-text NLG shared tasks (Gardent et al., 2017; Dušek et al., 2018; Castro Ferreira et al., 2020). Mapping the values as such can be sufficient for simple datasets, but otherwise, incorrect or incomplete delexicalisation will lead to inconsistent input and deteriorate performance.

Lastly, problems may also occur during relexicalisation as it does not take into account the context in which the placeholders are situated and may result in disfluent sentence. For a simplified example, observe how placing the unedited dates in the placeholders leads to disfluent output in Figure 2.

Segmentation strategies are commonly used in Neural Machine Translation to improve the generalisation ability of models. The objective is to break down words into smaller units, reducing the vocabulary and the number of unseen tokens (Sennrich et al., 2016). Unfortunately, applying segmentation in concept-to-text NLG, e.g. using Byte-Pair-Encoding (BPE) subword units (Gardent et al., 2017; Zhang et al., 2018) or using characters as basic units (Goyal et al., 2016; Agarwal and Dymetman, 2017; Deriu and Cieliebak, 2018), underperforms against delexicalisation. Challenges include capturing long dependencies between segmented words, and generating non-existing words.

Copy mechanism is another method to address unseen input, by allowing the decoder of an encoder-decoder model to draw a token directly from the input sequence instead of generating it from the decoder vocabulary (See et al., 2017). While applications of the copy mechanism in concept-to-text NLG have achieved overall good results (Chen, 2018; Elder et al., 2018; Gehrmann et al., 2018), when dealing with rare and unseen inputs delexicalisation is still preferable (Shimorina and Gardent, 2018). To improve the generalisation ability of copy mechanism models, Roberti et al.

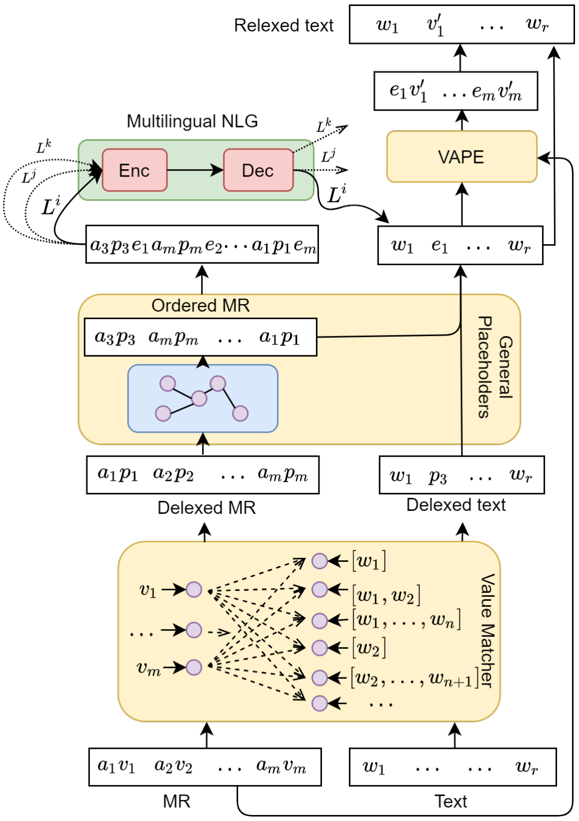


Figure 3: Language Agnostic Delexicalisation outline.

(2019) propose applying the copy mechanism to character-level NLG systems. This is combined with an additional optimisation phase during training where the encoder and decoder are switched.

4 Language Agnostic Delexicalisation

In order to address the shortcomings of previous approaches to generalise over rare or unseen inputs, especially in cases of multilingual output, we propose Language Agnostic Delexicalisation (LAD). Figure 3 shows an overview of LAD; the input and output are first delexicalised using pretrained language-independent embeddings, and (optionally) ordered. The multilingual generation model is trained on the delexicalised training data, and the output is relexicalised using automatic value post-editing to ensure that the values fit the context. Each component is described in more detail bellow.

To enable multilingual generation, we adapt the universal encoder-decoder framework via "*target forcing*" (Johnson et al., 2017) since it can be directly applied to any NLG model without the need to modify the latter's architecture. To do so, we extend the input MR in the encoder with a language token that signals which language the model should generate output in. In addition, we follow Wang et al. (2018) and initialise the decoder with the language token. The rest of the components (i.e. delexicalisation, ordering, and value post-editing) are orthogonal to the model's architecture.

4.1 Value Matching

As discussed in Section 3, one of the challenges of delexicalisation is matching the MR values with corresponding words in the text, especially in the multilingual setting. Even when the MR values are in the same language as the target, we observe from the examples in Figure 1 that token overlapping methods (i.e. exact and n-gram matching) are not sufficient to generate a complete and accurate delexicalisation as values may appear differently.

To counter this problem, LAD performs matching by mapping MR values to n-grams based on the similarity of their representations. Specifically, it calculates the similarity between a value v and all word n-grams $w_i \ldots w_j$ in the text, with $j - i < n$ and n set to the maximum value length observed in the training data. LAD employs LASER (Artetxe and Schwenk, 2019) to generate language agnostic sentence embeddings of the values and n-grams, and calculates their distance via cosine similarity. Given an MR and text, all possible value and n-gram comparisons are calculated and the matches are determined in a greedy fashion.

4.2 Generic Placeholders and Ordering

In Section 3, we discussed how the WebNLG datasets are more challenging because they contain unseen attributes in the development and test subsets, in addition to unseen values. This is problematic when we use attribute-bound placeholders (e.g. "X-type") as unseen attributes will result in unseen placeholders. Following Trisedya et al. (2018), for the WebNLG datasets, LAD uses numbered generic placeholders "X#" (e.g. "X1"). Unfortunately, the adoption of generic placeholders creates problems for relexicalisation as it becomes unclear which input value should replace which placeholder. We address this by ordering the model's input based on the graph formed by its RDF triples, again by following Trisedya et al. (2018). We traverse every edge in the graph, starting from the node with the least incoming edges (or randomly in case of ties) and then visit all nodes via BFS (breadth-first search). We then trust that the model will learn to respect the input order when generating, and follow the order to relexicalise the placeholders.

We note that this is only required for models that employ delexicalisation strategies and for datasets with unseen attributes (i.e. the WebNLG Challenge datasets). Concept-to-text NLG systems do not generally require ordered input (Wen et al., 2015).

4.3 Automatic Value Post-Editing

As discussed in Section 3, a naive relexicalisation of the placeholders may lead to disfluent sentences, as the procedure does not take into account the context in which the placeholders have been placed. For example, in the sentence "there are 2 X that have free parking", if we need to replace the placeholder "X" with the MR value "guesthouse" , the value should be pluralised to fit the context. This problem is more evident in morphologically rich languages, where more factors affect the value's form. To alleviate this, the LAD framework incorporates an Automatic Value Post-Editing component, consisting of a character-level seq2seq model that iterates over values as they are placed in the text and modifies them to fit the context of their respective placeholders. Anastasopoulos and Neubig (2019) has already shown the benefits of character models on morphological inflection generation, but no previous work has addressed how relexicalisation should adapt to context.

Our proposed VAPE model requires as input the MR placeholder e_i, original value v_i and corresponding NLG output $w'_1 \dots w'_n$ for context; these are serialised and passed to the encoder. Similar to the multilingual model, we add an appropriate language token L before the NLG output. The output of VAPE is the MR value v'_i in the proper form.

$$\{e_i\ v_i\ \text{[SEP]}\ L\ w'_1 \dots w'_n\} \rightarrow v'_i$$

The training signal for VAPE is obtained during delexicalisation. For a given delexicalisation strategy, we obtain all pairs of MR values and matching n-grams in the training data, and subsequently train VAPE using these n-grams as the targets. Therefore, the VAPE model is dependent on the quality of the delexicalisation strategy; specifically for exact delexicalisation, VAPE cannot be trained as the MR values and matching n-grams are the same.

Most edits VAPE performs concern incorrect inflections, but it is not limited to morphological edits and has the potential to deal with various types of modifications. During our experiments we observed VAPE performing value re-formatting (e.g. "1986_04_15" → *"April 15th 1986"*), synonym generation (e.g. *"east"* → *"oriental"*) and value translation (e.g. "bbc" from Latin to Cyrillic).

5 Experiments

For our experiments we use five datasets and calculate BLEU-4 (Papineni et al., 2002, ↑), METEOR (Banerjee and Lavie, 2005, ↑), chrF++ (Popović, 2015, ↑), and TER (Snover et al., 2006, ↓).

The WebNLG Challenge 2017 (Gardent et al., 2017, WebNLG17) data consists of sets of RDF triple pairs and corresponding English texts in 15 DBPedia categories. For our purposes, we will be using a later work (Shimorina et al., 2019) that introduced a machine translated Russian version of WebNLG17, a part of which was post edited by humans. Due to the limited amount of human corrected Russian sentences, and to facilitate the most accurate evaluation, we use these solely for testing. To ensure that half of the domains in the new test set remain unseen during training, we create our own train/dev/test split by retaining the following DBPedia categories from training and development sets: Astronaut, Monument and University.

The latest incarnation of the WebNLG Challenge (Castro Ferreira et al., 2020, WebNLG20) is fully human annotated for both English and Russian. We use this as the main dataset in our experiments, as it is designed to promote multilinguality. However, due to the fact that the provided test set does not contain unseen Russian instances, we perform our experiment on a custom split (WebNLG20*) ensuring that part of the domains in the test data remain unseen during training. The split was performed similarly to the previously described WebNLG17.

MultiWOZ 2.1 (Eric et al., 2020) and CrossWOZ (Zhu et al., 2020) are datasets of dialogue acts and corresponding utterances in English and Chinese respectively. The two datasets share the same structure, with MultiWOZ covering 7 domains and 25 attributes, and CrossWOZ covering 5 domains and 72 attributes; 4 of the domains are common in both datasets though CrossWOZ has more attached attributes. Multilingual WOZ 2.0 (Mrkšić et al., 2017) is also a dialogue dataset with utterances available in three languages: English, Italian and German. Its scope is more limited than MultiWOZ and CrossWOZ as it only covers a single domain.

For all models in our experiments, the input consists of a simple linearisation of the MRs. Particularly, for the delexicalisation based models, the values are extended with their respective placeholders as shown in the following example: "ENTITY_1 meyer werft *location* ENTITY_2 germany".

5.1 Ablation Study

First we perform an ablation study to determine how the different components of *LAD* (ordering and VAPE) affect its performance; *LAD* being our full Language Agnostic Delexicalisation model as described in Section 4. In addition to *LAD*, where these components are incrementally removed, we explore how their addition would influence exact and n-gram delexicalisation (Trisedya et al., 2018). We do not explore adding VAPE to exact delexicalisation (there is no *EXACT + O + V* variant), as it cannot be trained in this setting (see Section 4.3).

In Table 1, we observe that both components are beneficial, but less so for seen English data. For the more morphologically rich and lower resourced Russian, the components are helpful for both seen and unseen. VAPE leads to an improvement in performance in almost all cases and even when added on *NGram*. An exception is unseen English data, where removing VAPE is beneficial; this suggests that VAPE is overeager to make edits in English.

By studying the output, we observe that VAPE modified 20% of values in English, and 66% in Russian; directly copying the value was insufficient in Russian where proper inflection is needed. We identified three consistent errors where copying the original value would be preferable to using VAPE: the removal of date information (e.g. "1969-09-01 → 1st, 1969"), misspelling of proper nouns (e.g. "atatürk monument" → "atat erk monument"), and mishandling of long values (e.g. "ottoman army soldiers killed in the battle of baku" → "ottoman army soldiers killed in the batttle of kiled in the bathe batom"). We observe that these errors occur more frequently for English unseen cases, but could be reduced by extending VAPE with a control mechanism that decides whether copying the values themselves is preferable. Such errors occur in part because VAPE, as a character-level model, suffers from the same challenges as other segmentation methods (see Section 3). However, since VAPE's input is much shorter, the problem is not as prevalent. Overall, *LAD* outperforms the previous delexicalisation strategies *Exact* and *NGram*, and VAPE is shown to be integral to its performance.

	English			Russian		
	A	S	U	A	S	U
Exact	0.56	0.62	0.18	0.21	0.25	0.03
Exact + O	0.52	0.54	0.38	0.19	0.20	0.14
NGram	0.56	0.62	0.16	0.22	0.27	0.04
NGram + O	0.53	0.55	**0.40**	0.23	0.25	0.15
NGram + O+V	0.54	0.57	0.33	0.31	0.35	0.16
LAD - O-V	0.59	0.65	0.18	0.23	0.28	0.03
LAD - V	0.60	0.63	0.39	0.24	0.26	0.16
LAD	**0.62**	**0.66**	0.32	**0.37**	**0.42**	**0.21**

Table 1: BLEU on WebNLG20* for delexicalisation models augmented with generic placeholders+ordering, and value post-edit. A = All categories; S = Seen categories; U = Unseen categories; O = generic placeholders+ordering; V = Value post-edit.

		WOZ 2.0			WebNLG 2020*		MultiWOZ + CrossWOZ	
		en	it	de	en	ru	en	zh
Word	Mono	**0.66**	0.56	0.59	**0.57**	0.31	0.56	**0.68**
	Multi	0.65	**0.57**	**0.61**	**0.57**	**0.33**	**0.57**	0.66
LAD	Mono	0.66	0.58	0.56	0.58	0.32	0.56	**0.68**
	Multi	**0.68**	**0.59**	**0.57**	**0.62**	**0.37**	**0.58**	**0.68**

Table 2: BLEU for mono- and multilingual models.

5.2 Monolingual vs Multilingual

Here we explore the performance of monolingual and multilingual models on concept-to-text datasets. The *Word* model has the exact same architecture as *LAD* but no delexicalisation is performed, and consequently no automatic value post-editing and no ordering. Since there is no relexicalisation that needs to occur during post-processing, the input to the *Word* model needs not be specifically ordered, and is just a concatenation of the RDF triples as they appear in the original dataset. For multilingual, we add the appropriate language tokens on the input of *Word*, in the same manner we added them to *LAD*. For the monolingual (*Mono*) configuration we train the models to produce a single language, while for multilingual (*Multi*) we train them to produce all languages available in that dataset. Please refer to Table 2 for the results.

We observe that the multilingual models outperform their monolingual counterpart in most datasets and languages, especially with *LAD* as its delexicalisation and relexicalisation modules are more robust to multilingual input and output. Specifically for the MultiWOZ and CrossWOZ datasets, in the monolingual setting the models are trained exclusively on the respective dataset, i.e. MultiWOZ for English, and CrossWOZ for Chinese. For multilingual, we take advantage of the fact that these datasets share the same structure, and train the models on both datasets. For English,

English	All Categories				Seen Categories				Unseen Categories			
	BLEU	METEOR	chrF++	TER	BLEU	METEOR	chrF++	TER	BLEU	METEOR	chrF++	TER
Word	0.57	0.36	0.63	0.44	0.64	0.43	0.72	0.36	0.10	0.11	0.26	0.90
Char	0.54	0.35	0.52	0.47	0.61	0.41	0.70	0.40	0.05	0.09	0.24	0.89
BPE	0.54	0.35	0.63	0.49	0.62	0.42	0.72	0.42	0.07	0.09	0.24	0.97
SP	0.58	0.37	0.64	0.42	**0.66**	0.44	0.74	0.34	0.07	0.09	0.23	0.96
Copy	0.57	0.36	0.63	0.45	0.59	0.38	0.65	0.42	**0.38**	0.27	0.51	**0.61**
LAD	**0.62**	**0.42**	**0.71**	**0.71**	**0.66**	**0.45**	**0.75**	**0.31**	0.32	**0.30**	**0.54**	**0.61**

Russian	All Categories				Seen Categories				Unseen Categories			
	BLEU	METEOR	chrF++	TER	BLEU	METEOR	chrF++	TER	BLEU	METEOR	chrF++	TER
Word	0.33	0.41	0.44	0.63	0.42	0.51	0.54	0.56	0.02	0.13	0.20	0.94
Char	0.30	0.41	0.44	0.67	0.38	0.50	0.53	0.61	0.01	0.13	0.20	0.91
BPE	0.25	0.38	0.44	0.71	0.32	0.48	0.54	0.65	0.02	0.11	0.19	0.97
SP	0.34	0.41	0.45	0.62	**0.44**	0.54	0.56	0.53	0.01	0.10	0.18	0.98
Copy	0.24	0.35	0.39	0.74	0.29	0.42	0.46	0.72	0.02	0.14	0.20	0.91
LAD	**0.37**	**0.51**	**0.55**	**0.55**	0.42	**0.57**	**0.60**	**0.51**	**0.21**	**0.34**	**0.42**	**0.71**

Table 3: WebNLG20* results for Multilingual models.

English	All Categories				Seen Categories				Unseen Categories			
	BLEU	METEOR	chrF++	TER	BLEU	METEOR	chrF++	TER	BLEU	METEOR	chrF++	TER
SP	0.58	0.37	0.64	0.42	0.66	0.44	0.74	0.34	0.07	0.09	0.23	0.96
LAD	0.62	0.42	0.71	0.36	0.66	0.45	**0.75**	0.31	0.32	0.30	0.54	0.61
mBART	0.66	0.44	0.74	0.33	0.67	0.45	**0.75**	0.32	0.58	0.41	0.70	0.44
mB-LAD	0.66	0.44	0.74	**0.31**	**0.68**	**0.46**	**0.75**	**0.30**	0.52	0.38	0.68	0.44
mB-LAD+	**0.67**	**0.45**	**0.75**	**0.31**	**0.68**	**0.46**	**0.75**	**0.30**	**0.61**	**0.42**	**0.71**	**0.37**
mB-LAD-SPE	0.66	0.44	0.74	**0.31**	0.67	0.45	**0.75**	**0.30**	0.59	0.41	0.70	0.38

Russian	All Categories				Seen Categories				Unseen Categories			
	BLEU	METEOR	chrF++	TER	BLEU	METEOR	chrF++	TER	BLEU	METEOR	chrF++	TER
SP	0.34	0.41	0.45	0.62	0.44	0.54	0.56	0.53	0.01	0.10	0.18	0.98
LAD	0.37	0.51	0.55	0.55	0.42	0.57	0.60	0.51	0.21	0.34	0.42	0.71
mBART	0.37	0.50	0.51	0.57	0.43	0.56	0.57	0.52	0.15	0.33	0.35	0.78
mB-LAD	0.41	0.54	0.58	0.51	**0.45**	**0.58**	**0.61**	**0.49**	0.29	0.44	0.48	0.59
mB-LAD+	0.42	0.55	0.58	0.51	0.41	0.57	0.60	0.50	0.41	**0.52**	**0.55**	0.52
mB-LAD-SPE	**0.46**	**0.57**	**0.59**	**0.47**	0.42	0.57	0.60	**0.49**	**0.44**	**0.52**	**0.55**	**0.49**

Table 4: WebNLG20* results for Pretrained Multilingual models.

we observe that the multilingual model improves, suggesting that domain knowledge is transferred from CrossWOZ. For Chinese however, the multilingual *Word* model underperforms. This is not very surprising, as the overlap between the datasets is favourable to MultiWOZ, i.e. most of the attributes of MultiWOZ also appear in CrossWOZ, while the majority of CrossWOZ's attributes do not appear in MultiWOZ.

5.3 Multilingual Generalisation

Tables 3 contains full results for English and Russian on WebNLG20* respectively. We include the *Word* configuration (see Section 5.2), as well as *Char*, *BPE*, and *SP*, which are variations that use characters, Byte-Pair-Encoding, and SentencePiece as subword units respectively. *Copy* refers to the copy mechanism model by Roberti et al. (2019). The *SP* model performs very well for seen categories, but fails to generalise on unseen data. The *Copy* model performs well for unseen categories in English, but underperforms in Russian as values for it are only partially translated, i.e. some values in the MR may appear in English while others appear in Russian. This is challenging for *Copy* models as the target reference does not closely match the input, but *LAD* can handle it more robustly.

Observing the output, *LAD*'s main advantage is that it avoids under- and over-generating values as they are being controlled by the placeholders.[1] *SP* is often the most fluent of the models, but for

[1] We provide output examples in the Appendix.

English	All Categories				Seen Categories				Unseen Categories			
	BLEU	METEOR	chrF++	TER	BLEU	METEOR	chrF++	TER	BLEU	METEOR	chrF++	TER
mBART	0.49	0.37	0.63	0.46	0.55	0.40	0.68	0.42	0.41	0.33	0.57	0.50
mB-LAD	0.49	**0.39**	0.67	0.44	**0.56**	**0.41**	**0.71**	**0.41**	0.40	0.36	0.62	0.48
mB-LAD+	**0.50**	**0.39**	**0.68**	**0.43**	0.55	**0.41**	**0.71**	**0.41**	**0.44**	**0.38**	**0.64**	**0.46**
mB-LAD-SL	0.48	**0.39**	0.66	0.45	0.54	**0.41**	0.70	**0.41**	0.40	0.36	0.61	0.49

Table 5: Official WebNLG20 testset results for Pretrained Multilingual models on English text.

longer input it tends to under-generate and miss values. The *Copy* model tends to repeat values, which can be attributed to the fact that it is based on characters where long-distance dependencies are hard to maintain. On the other hand, *Copy* can potentially generate more relevant output since it can copy words from attributes as well as values.

Overall, *LAD* helps the multilingual model outperform all other models in both English and Russian. It is especially beneficial in generalising to unseen data, as was its main objective after all.

5.4 Generalising with Pretrained Models

Here we explore the generalisation capabilities of multilingual pretrained models, by replacing the underlying NLG model with mBART (Liu et al., 2020), a multilingual denoising autoencoder pretrained on a large-scale dataset containing 25 languages (CC25). Similarly to Kasner and Dušek (2020), we fine-tune mBART with the default EN-RO configuration for up to 10000 updates. Using mBART as the underlying model also helps facilitate a comparison against a configuration that is similar to many of the state of the art participants in the WebNLG 2020 Challenge, although some of them used different pretrained models.

Table 4 shows the performance of the fine-tuned models on the WebNLG20* dataset. The mBART-based model outperforms the non-delexicalisation *SP*, and non-pretrained *LAD* in English. However, *LAD* still performs better in Russian. This makes sense as the CC25 dataset is heavily biased towards the English language and contains double the amount of tokens compared to Russian, and much more compared to other lower-resource languages. Combining the LAD framework with mBART (*mB-LAD*) resulted in a general improvement in performance, especially for lower-resource unseen data. However, as discussed in Section 5.1, the VAPE component remains to some degrees susceptible to unseen contexts. To tackle this issue, we improve VAPE by pre-loading mBART and fine-tuning it for value post-editing as well (*mB-LAD+*), achieving 3 and 29 points increase in BLEU score for unseen English over the vanilla *mBART* and *LAD* models, and 26 and 20 points for unseen Russian. Additionally, to take advantage of mBART's denoising ability, we extend the fine-tuned VAPE to edit the "exact" relexicalised NLG output and provide a sentence-level output (*mB-LAD-SPE*), i.e. edits are not exclusively focused on the values. Results show that *mB-LAD-SPE* improves further *mB-LAD+* on Russian in both seen and unseen.

Russian	All/Seen Categories			
	BLEU	METEOR	chrF++	TER
mBART	0.43	0.55	0.56	0.52
mB-LAD	0.42	**0.61**	**0.64**	0.50
mB-LAD+	0.38	0.59	0.61	0.53
mB-LAD-SL	**0.44**	**0.61**	0.63	**0.48**

Table 6: Official WebNLG20 testset results for Pretrained Multilingual models on Russian text.

Russian	All Categories			
	BLEU	METEOR	chrF++	TER
Word	0.02	0.16	0.21	0.95
Char	0.01	0.14	0.19	0.90
BPE	0.02	0.15	0.20	0.97
SP	0.02	0.16	0.20	0.93
LAD	**0.04**	**0.22**	**0.26**	**0.84**

Table 7: WebNLG17 results for Multilingual models.

Table 5 and 6 also shows the automatic evaluation of the fine-tuned mBART models on the official WebNLG20 Challenge testset; the official test set had no unseen subset of Russian. The results are consistent with the findings in our previous experiments, with small improvements of LAD-based mBART models over the mBART-base.

5.5 Synthetic Data

We use the WebNLG17 automatically translated Russian "silver" data, to determine how useful they are for training multilingual concept-to-text NLG. As preliminary results were not promising, we limit the scope of the experiment to only a few systems. Table 7 gathers the results It is apparent that automatically translated data are insufficient; *LAD* seems to more consistently achieve higher performance than other models, but all scores are too low to draw any sufficiently supported conclusions.

6 Conclusion

We proposed Language Agnostic Delexicalisation, a novel delexicalisation framework that matches and delexicalises MR values in the text independently of the language. For relexicalisation, an automatic value post editing model adapts the values to their context. Results show that multilingual models outperform monolingual models, and that LAD outperforms previous work in improving the performance of multilingual models, especially in low resource conditions. LAD also improves on the performance of pre-trained language models achieving state-of-the-art results. The automatic value post editing component is especially beneficial in morphologically rich languages.

References

Shubham Agarwal and Marc Dymetman. 2017. A surprisingly effective out-of-the-box char2char model on the E2E NLG challenge dataset. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 158–163, Saarbrücken, Germany. Association for Computational Linguistics.

Antonios Anastasopoulos and Graham Neubig. 2019. Pushing the limits of low-resource morphological inflection. 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 984–996, Hong Kong, China. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

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.

Muthu Kumar Chandrasekaran, Michihiro Yasunaga, Dragomir R. Radev, Dayne Freitag, and Min-Yen Kan. 2019. Overview and results: Cl-scisumm shared task 2019. In *Proceedings of the 4th Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2019) co-located with the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019), Paris, France, July 25, 2019*.

Shuang Chen. 2018. A general model for neural text generation from structured data. *E2E NLG Challenge System Descriptions*.

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2020. Cross-lingual natural language generation via pre-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7570–7577.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Jan Milan Deriu and Mark Cieliebak. 2018. End-to-end trainable system for enhancing diversity in natural language generation. *E2E NLG Challenge System Descriptions*.

Xiangyu Duan, Mingming Yin, Min Zhang, Boxing Chen, and Weihua Luo. 2019. Zero-shot cross-lingual abstractive sentence summarization through teaching generation and attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3162–3172, Florence, Italy. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 322–328, Tilburg University, The Netherlands. Association for Computational Linguistics.

Henry Elder, Sebastian Gehrmann, Alexander O'Connor, and Qun Liu. 2018. E2E NLG challenge submission: Towards controllable generation of diverse natural language. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 457–462, Tilburg University, The Netherlands. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Angela Fan and Claire Gardent. 2020. Multilingual AMR-to-text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2889–2901, Online. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

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

Sebastian Gehrmann, Falcon Dai, Henry Elder, and Alexander Rush. 2018. End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 46–56, Tilburg University, The Netherlands. Association for Computational Linguistics.

Raghav Goyal, Marc Dymetman, and Eric Gaussier. 2016. Natural language generation through character-based RNNs with finite-state prior knowledge. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1083–1092, Osaka, Japan. The COLING 2016 Organizing Committee.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudoh. 2019. Findings of the third workshop on neural generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 1–14, Hong Kong. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Zdeněk Kasner and Ondřej Dušek. 2020. Train hard, finetune easy: Multilingual denoising for RDF-to-text generation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 171–176, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. 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.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

Lesly Miculicich, Marc Marone, and Hany Hassan. 2019. Selecting, planning, and rewriting: A modular approach for data-to-document generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 289–296, Hong Kong. Association for Computational Linguistics.

Simon Mille, Anya Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham, and Leo Wanner. 2020. The third multilingual surface realisation shared task (SR'20): Overview and evaluation results. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 1–20, Barcelona, Spain (Online). Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. 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.

Ratish Puduppully, Jonathan Mallinson, and Mirella Lapata. 2019. University of Edinburgh's submission to the document-level generation and translation shared task. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 268–272, Hong Kong. Association for Computational Linguistics.

Marco Roberti, Giovanni Bonetta, Rossella Cancelliere, and Patrick Gallinari. 2019. Copy mechanism and tailored training for character-based data-to-text generation. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part II*, pages 648–664.

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.

Shi-qi Shen, Yun Chen, Cheng Yang, Zhi-yuan Liu, and Mao-song Sun. 2018. Zero-shot cross-lingual neural headline generation. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 26(12):2319–2327.

Anastasia Shimorina and Claire Gardent. 2018. Handling rare items in data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 360–370, Tilburg University, The Netherlands. Association for Computational Linguistics.

Anastasia Shimorina, Elena Khasanova, and Claire Gardent. 2019. Creating a corpus for Russian data-to-text generation using neural machine translation and post-editing. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 44–49, Florence, Italy. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, and Wei Wang. 2018. GTR-LSTM: A triple encoder for sentence generation from RDF data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1627–1637, Melbourne, Australia. Association for Computational Linguistics.

Xiaojun Wan, Huiying Li, and Jianguo Xiao. 2010. Cross-language document summarization based on machine translation quality prediction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 917–926, Uppsala, Sweden. Association for Computational Linguistics.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2955–2960, Brussels, Belgium. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Biao Zhang, Jing Yang, Qian Lin, and Jinsong Su. 2018. Attention regularized sequence-to-sequence learning for e2e nlg challenge. *E2E NLG Challenge System Descriptions*.

Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. CrossWOZ: A large-scale Chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*, 8:281–295.

7 Appendices

A Configurations

The multilingual NLG and VAPE use a transformer as underlying architecture. We use the fairseq toolkit for our experiments (Ott et al., 2019). The models are trained with shared embeddings, 8 attention heads, 6 layers, 512 hidden size, 2048 size for the feed forward layers. We trained with 0.3 dropout, adam optimiser with a learning rate of 0.0005. The NLG are trained with early stopping and patience set to 20. Automatic value post edit models are trained with the same configuration but patience was set to 6. For the copy mechanism-based model we use the EDA-CS implementation provided by Roberti et al. (2019) with the default configuration. Due to its extremely high computational training cost, the models are trained for 15 epochs. BPE and SentencePiece (Kudo and Richardson, 2018) models are trained with a vocabulary size set to 12000 tokens.

For all models in our experiments, the input consists of a simple linearisation of the MRs. Particularly, for the delexicalisation based models, the values are extended with their respective placeholders as shown in the following example: "ENTITY_1 meyer werft *location* ENTITY_2 germany.

B Input examples

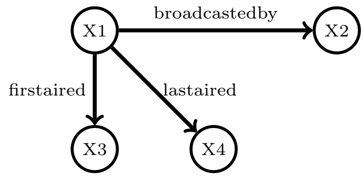
Figure 6 shows some examples of how, during training, LAD maps MR values to n-grams of the target reference, based on the similarity of their representations. We can observe that these values could not have been matched by exact and n-gram delexicalisation as they constitute significant paraphrases of the value.

Figure 4 and 5 show some additional examples of delexicalisation and relexialisation for the various approaches from the WebNLG Challenge 2020. Table 8 shows more delexicalisation examples from WebNLG, MultiWOZ and CrossWOZ datasets, where we can observe the shortcomings of exact and n-gram delexicalisation.

C Output examples

Table 9 and 10 present some examples for English and Russian output respectively. The examples include output from SentencePiece (SP), Copy, and LAD systems.

Input MR:



where: **X1**=bananaman|бананамен, **X2**=bbc|би_би_си, **X3**=1983_10_03, **X4**=1986_04_15

Gold Target References:

bananaman first aired on the **bbc** on **october 3rd, 1983** and broadcast its last episode on **april 15th, 1986**.

бананамен впервые вышел в эфир на **bbc 3 октября 1983** года, а его последний эпизод вышел **15 апреля 1986** года.

Exact Delexicalisation:

X1 first aired on the **X2** on october 3rd, 1983 and broadcast its last episode on april 15th, 1986.

X1 впервые вышел в эфир на bbc 3 октября 1983 года, а его последний эпизод вышел 15 апреля 1986 года.

Language Agnostic Delexicalisation (LAD):

X1 first aired on the **X2** on october **X3** and broadcast its last episode on **X4**.

X1 впервые вышел в эфир на **X2 X3** года, а его последний эпизод вышел **X4** года.

Figure 4: Delexicalisation on WebNLG Challenge 2020 with target output in English and Russian. Double underlining marks text missed by delexicalisation.

Generated Output before Relexicalisation:
(assuming training with LAD)

X1 first aired on the **X2** on october **X3** and broadcast its last episode on **X4**.

X1 впервые вышел в эфир на **X2 X3** года, а его последний эпизод вышел **X4** года.

Exact Relexicalisation:

bananaman first aired on the **bbc** on october **1983 10 03** and broadcast its last episode on **1986 04 15**.

бананамен впервые вышел в эфир на **би би си 1983 10 03** года, а его последний эпизод вышел **1986 04 15** года.

Automatic Value Post-Editing (VAPE):

bananaman first aired on the **bbc** on october **3rd, 1983** and broadcast its last episode on **april 15th, 1986**.

бананамен впервые вышел в эфир на **бибиси 3 октября 1983** года, а его последний эпизод вышел **15 апреля 1986** года.

Figure 5: Relexicalisation examples; double underlining marks errors that ignore context.

Example taken from the WebNLG 2020 dataset.	
MR	⟨**X1**=bananaman, broadcastedby, **X2**=bbc⟩ ⟨**X1**=бананамен, broadcastedby, **X2**=би_би_си⟩ ⟨**X1**=bananaman, firstaired, **X3**=1983_10_03⟩ ⟨**X1**=бананамен, firstaired, **X3**=1983_10_03⟩ ⟨**X1**=bananaman, lastaired, **X3**=1986_04_15⟩ ⟨**X1**=бананамен, lastaired, **X4**=1986_04_15⟩
Reference	bananaman first aired on the bbc on october 3rd , 1983 and broadcast its last episode on april 15th , 1986. бананамен впервые вышел в эфир на bbc 3 октября 1983 года , а его последний эпизод вышел 15 апреля 1986 года .
Exact	**X1** first aired on the **X2** on october 3rd , 1983 and broadcast its last episode on april 15th , 1986. **X1** впервые вышел в эфир на bbc 3 октября 1983 года , а его последний эпизод вышел 15 апреля 1986 года .
NGram	**X1** first aired on the **X2** on october 3rd , **X3** and broadcast its last episode on april 15th **X4**. **X1** впервые вышел в эфир 3 октября **X3** года , а его последний эпизод вышел 15 апреля **X4** года .
LAD	**X1** first aired on the **X2** on **X3** and broadcast its last episode on **X4**. **X1** впервые вышел в эфир на **X2** **X3** года , а его последний эпизод вышел **X4** года .

Example taken from the MultiWOZ dataset.	

MR	hotel-inform{type:**X1**="guesthouse", parking:none, choice:**X2**="5"}
	booking-inform{none}

Reference	there are 5 guesthouses that have free parking . should i book one of them for you ?
Exact	there are **X2** guesthouses that have free parking . should i book one of them for you ?
NGram	there are **X2** guesthouses that have free parking . should i book one of them for you ?
LAD	there are **X2** **X1** that have free parking . should i book one of them for you ?

Example taken from the CrossWOZ dataset.	

MR	attraction-inform{duration:**X1**="1 小时", rating:**X2**= "5 分"}
	attraction-request{name:none}

Reference	吃完饭，我想去一个评分 5 分的景点，转上 1 个来小时，你能给我推荐一个吗
Exact	吃完饭，我想去一个评分 **X2** 的景点，转上 1 个来 小时 ，你能给我推荐一个吗
NGram	吃完饭，我想去一个评分 **X2** 的景点，转上 1 个来 **X1** ，你能给我推荐一个吗
LAD	吃完饭，我想去一个评分 **X2** 的景点，转上 **X1** ，你能给我推荐一个吗

Table 8: Dataset examples and delexicalisation output; double underlining marks text that was missed.

Target:	aarhus airport is located in tirstrup , part of the central region of denmark which has the capital city of copenhagen .
Value:	central denmark region
Cos	*n-gram*
0.95	**the central region of denmark**
...	...
0.73	the capital city of copenhagen
...	...
0.25	which has the

Target:	alan shepard is dead .
Value:	deceased
Cos	*n-gram*
0.84	**dead**
...	...
0.47	alan shepard
...	...
0.40	alan shepard is dead .

Figure 6: Examples of LAD's value mapping to target reference n-grams.

MR: ⟨ Trane, revenue, 1.0264E10 ⟩ ⟨ Trane, netIncome, 5.563E8 ⟩ ⟨ Trane, numberOfEmployees, 29000 ⟩
SP: trane has a revenue of $ 10,264,000,000 , with a net income of $ 556,300,000 and a revenue of $ 10,264,000,000 .
Copy: trane , a company with 29,000 employees , has 29,000 employees and was connected at $ 556,300,000 .
LAD: trane , which has a revenue of $ 10,264,000,000 , has a net income of $ 556,300,000 and employs 29,000 people .
MR: ⟨ William_Anders, dateOfRetirement, "1969-09-01"⟩ ⟨ William_Anders, occupation, Fighter_pilot ⟩ ⟨ William_Anders, birthPlace, British_Hong_Kong ⟩ ⟨ William_Anders, was a crew member of, Apollo_8 ⟩
SP: the birth place of greek born , adonis georgiadis , is the company , of which was in office at the same time that m ogenenenenenenenenville , new britain , connecticut , is a member of the order of poales and a division of 45000 kilometres .
Copy: william anders was born in british hong kong and has a crew mew member of the fighter pilot . the was a crew member of the was a crew member of the was a crew member of the was a crew member of
LAD: william anders , which was followed by 1st , 1969 and fighter pilot , was born in british hong kong and has been a number of apollo 8 .

Table 9: Output text from three different systems in English.

MR: ⟨ Trane, revenue, 1.0264E10 ⟩ ⟨ Trane, netIncome, 5.563E8 ⟩ ⟨ Trane, numberOfEmployees, 29000 ⟩
SP: trane has a revenue of $ 10,264,000,000 , with a net income of $ 556,300,000 and a revenue of $ 10,264,000,000 .
Copy: trane , a company with 29,000 employees , has 29,000 employees and was connected at $ 556,300,000 .
LAD: trane , which has a revenue of $ 10,264,000,000 , has a net income of $ 556,300,000 and employs 29,000 people .
MR: ⟨ William_Anders, dateOfRetirement, "1969-09-01"⟩ ⟨ William_Anders, occupation, Fighter_pilot ⟩ ⟨ William_Anders, birthPlace, British_Hong_Kong ⟩ ⟨ William_Anders, was a crew member of, Apollo_8 ⟩
SP: the birth place of greek born , adonis georgiadis , is the company , of which was in office at the same time that m ogenenenenenenenenville , new britain , connecticut , is a member of the order of poales and a division of 45000 kilometres .
Copy: william anders was born in british hong kong and has a crew mew member of the fighter pilot . the was a crew member of the was a crew member of the was a crew member of the was a crew member of
LAD: william anders , which was followed by 1st , 1969 and fighter pilot , was born in british hong kong and has been a number of apollo 8 .

Table 10: Output text from three different systems in Russian.