WebNLG Challenge 2020: Language Agnostic Delexicalisation for Multilingual RDF-to-text generation

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

This paper presents our submission to the WebNLG Challenge 2020 for the English and Russian RDF-to-text generation tasks. Our first of three submissions is based on Language Agnostic Delexicalisation, a novel delexicalisation method that match values in the input to their occurrences in the corresponding text through comparison of pretrained multilingual embeddings, and employs a character-level post-editing model to inflect words in their correct form during relexicalisation. Our second submission forfeits delexicalisation and uses SentencePiece subwords as basic units. Our third submission combines the previous two by alternating between the output of the delexicalisation-based system when the input contains unseen entities and/or properties and the output of the SentencePiece-based system when the input is seen during training.

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). It is common practice to perform a *delexicalisation* (Wen et al., 2015) of the MR, in order to facilitate the NLG model's generalization 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 is a two-step process that starts with a preprocessing step where all occurrences of MR values in the text are replaced with placeholders. This way the model focuses on learning to generate text that is abstracted away from the actual values. In a post-processing step, known as relexicalisation, the placeholders are re-filled with the MR values. Delexicalization does not need to be contained to solely the MR values; in the Surface Realization task, full delexicalization of the input structures has also been explored to great effect (Shimorina and Gardent, 2019; Colin and Gardent, 2019).

The main shortcoming of delexicalisation is that its efficacy is bound by the number of values that are correctly identified. In fact, a naive implementation of delexicalisation 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 context-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 that does not consider how the context should effect the morphology of the MR value when it is added to the text and vise versa (Goyal et al., 2016).

For our submissions to the WebNLG Challange 2020 RDF-to-text generation tasks, we trained two multilingual neural NLG models that use differently-preprocessed data. Our first submitted system is based on 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

3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+), Dublin, Ireland (Virtual), 18 December 2020, pages 186–191,

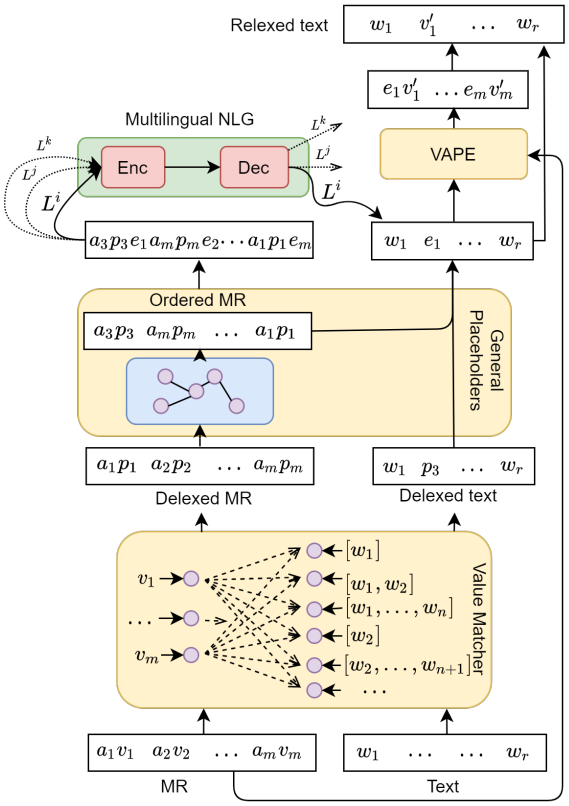

Figure 1: Language Agnostic Delexicalisation outline

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. Our second submission does not employ delexicalisation but makes use of an additional segmentation step that breaks down words into subword units. We anticipate LAD to perform better for inputs that were not seen during training (due to it abstracting away from specific values) and the subword-based model to perform better for seen inputs. To combine the two models, we submitted a third system that simply selects the output of the LAD system when the input contains unseen entities and/or properties or the output of the subword-based system when the input is seen.

2 Language Agnostic Delexicalisation

In this section, we will describe the Language Agnostic Delexicalisation (LAD) framework employed by our first submitted system. Figure 1 shows an overview of LAD; the input and output are first delexicalised using pretrained language-independent embeddings, and ordered. The multilingual generation model is trained on the delexicalised 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.

2.1 Value Matching

As briefly discussed in the introduction, 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 instances in the dataset that token overlapping methods are not sufficient to generate a complete and accurate delexicalisation as values may appear differently in the text.

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 $w_i \dots w_j$ 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.

2.2 Generic placeholders and ordering

The WebNLG dataset is challenging as it contains properties unseen during training, in addition to unseen values. This is problematic when we use property-bounded placeholders (e.g. "AIRPORT") as unseen properties will result in unseen placeholders. Following Trisedya et al. (2018), LAD uses numbered generic placeholders "ENTITY-#" (e.g. "ENTITY-1"). 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.

2.3 Automatic Value Post-Editing

As previously mentioned, 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. 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 (VAPE) component, consisting of a character-level seq2seq model that iterates over the values as they are placed in the text and modifies them to fit the context of their respective placeholders. Previous work (Anastasopoulos and Neubig, 2019) has already established the effectiveness 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.

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).

Lastly, to counter the overprocessing of the values, we employ the same strategy used in Section 2.1 to measure the similarity between the original and post-processed value and discard the modifications when the similarity score does not reach a predefined threshold (i.e. the relexicalisation is performed using the original value).

2.4 Automatic Results

3 Data and preprocessing

In the WebNLG Challenge RDF-to-text task, the goal is to generate text that describes particular entities and their properties as they were extracted from Knowledge Bases, and as such the input MR consists of a set of RDF (Resource Description Format) triples, each in the form of <subject, property, object>.

Our submitted systems are trained solely on the data provided by the organiser of the challenge. The MRs are formed by concatenating the triples in the RDFs sequentially. Tokenisation and truecasing is performed with the scripts provided in the Moses toolkit (Koehn et al., 2007). 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 <2lang> 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 as well.

For the LAD-based system, both MR and output text are delexicalised. During training, the input MR is ordered by imitating the order in which the triple's objects appear in the corresponding text (according to LAD-based matching). During testing, we follow Trisedya et al. (2018) and perform graph-based ordering. The values in the input are delexicalized and indicated with a property-related placeholder followed by a numbered general placeholder. In the output text, the values are replaced solely with the numbered general placeholders. Examples of inputs and outputs are shown in Table 1. The subword-based system of our secondary submission uses SentencePiece (Kudo and Richardson, 2018) as segmentation strategy. We train the model with the implementation provided by Wolf et al. (2019) and vocabulary size set to 8000.

4 Configurations

The multilingual NLG and VAPE use a transformer as underlying architecture. We use the fairseq toolkit for our systems (Ott et al., 2019). Hyperparameters are fine-tuned for the LAD system via random search on the validation set and corpus BLEU as scoring function. The resulting model

RDF	<Barny Cakes | dishVariation | Apple> <Barny Cakes | carbohydrate | 18.0 g> <Barny Cakes | protein | 1.8 g>
LAD input	<2en> D_Food FOOD ENTITY_1 dishVariation DISHVARIATION ENTITY_2 FOOD ENTITY_1 protein PROTEIN ENTITY_3 FOOD ENTITY_1 carbohydrate CARBOHYDRATE ENTITY_4
SP input	_<2en> _D_Food _Barny _cakes _dishvariation _ap ple _Barny _cakes _carbohydr ate _18 .0 _g _Barny _cakes _protein _1 .8 _g
Original Text	Barny cakes can be made with apple and contain 1.8g protein and 18g of carbs.
LAD output	<2en> ENTITY_1 can be made with ENTITY_2 and contain ENTITY_3 protein and ENTITY_4 of carbs .
SP target	_<2en> _Barny _cakes _can _be _made _with _ap ple _and _contain _1 .8 g _protein _and _18 g _of _carb s _.

Table 1: Example of preprocessed input and output for the LAD-based and subword SentencePiece (SP)-based systems.

En-All	BLEU	METEOR	CHRF++	TER	BERT PRECISION	BERT RECALL	BERT F1
LAD	39.55	**0.372**	**0.613**	0.536	0.935	**0.937**	**0.935**
SP	24.45	0.367	0.608	**0.522**	**0.936**	0.935	**0.935**
LAD+SP	**41.03**	0.223	0.425	0.739	0.874	0.880	0.876
En-Seen							
LAD	49.68	**0.402**	**0.674**	0.504	0.950	**0.949**	**0.949**
SP	51.85	0.383	0.651	0.464	0.954	0.945	**0.949**
LAD+SP	**52.93**	0.391	0.661	**0.457**	**0.955**	0.945	**0.949**
En-Unseen-Cat							
LAD	29.13	**0.345**	**0.553**	**0.575**	**0.922**	**0.926**	**0.924**
SP	7.87	0.138	0.288	0.851	0.841	0.855	0.847
LAD+SP	**29.39**	0.325	0.523	0.693	0.908	0.909	0.908
En-Unseen-Ent							
LAD	**42.42**	**0.375**	**0.631**	**0.487**	**0.944**	**0.944**	**0.944**
SP	10.82	0.166	0.335	0.825	0.848	0.855	0.851
LAD+SP	**42.42**	**0.375**	**0.631**	**0.487**	**0.944**	**0.944**	**0.944**
Russian							
LAD	24.87	0.523	0.537	0.673	0.849	0.855	0.851
SP	**46.84**	**0.632**	**0.637**	**0.456**	**0.899**	**0.890**	**0.893**
LAD+SP	41.52	0.602	0.610	0.486	0.891	0.883	0.886

Table 2: Automatic evaluation on testset for the English (all, seen categories, unseen categories, unseen entity) and Russian tasks.

is trained with shared embeddings, 2 encoder and 1 decoder attention heads, 4 layers, 256 hidden size, 3072 size for the feed forward layers. We trained with 0.4 dropout and 0.1 attention dropout, adam optimiser with a learning rate of 3e-4, clip normalisation of 2, early stopping and patience set to 20. The subword system and the VAPE model are trained with the same configuration.

The VAPE overprocessing similarity check threshold was set to 0.95.

5 Evaluation

Table 2 shows the automatic evaluation results for the submitted systems, as they were provided by the shared task (Castro-Ferreira et al., 2020).

We observe that overall for English *LAD* performs better than the subword model *SP*. Both *LAD* and *SP* performed comparably on instances seen at training time. However, the latter underperforms when dealing with unseen entities and even worse with unseen categories, i.e. domains not seen during training that often contain unseen properties in

addition to unseen entities. Surprisingly, the combination system *LAD+SP* performs better than *SP* in seen inputs and better than *LAD* in unseen categories. This implies some inconsistency between how *LAD+SP* distinguishes seen from unseen inputs and how they were divided for the shared task.

On the Russian test set, the subword system outperformed LAD considerably across all the evaluation metrics. We note however that the Russian test set contained practically no unseen properties nor unseen entities. This is an unfavourable setting for LAD as it is specifically designed to address generalisation to unseen input. Some properties do appear inconsistently in testing (e.g. spelled differently from how they occurred in training) which explains why the combination system *LAD+SP* performs worse than single *SP*.

By directly observing the output, we note that *LAD* is less fluent than the non-delexicalisation model, as the generated context is not always consistent with the values. However, its main advantage is that it avoids under- and over-generating values as they are being controlled by the placeholders. In fact, while *SP* often appears as more fluent than LAD, it tends to under-generate and miss values, especially for longer inputs.

6 Conclusion

We presented our submission to the WebNLG Challenge 2020 English and Russian RDF-to-text generation tasks. For both tasks we employ multilingual transformer-based seq2seq NLG systems that differ, however, in the type of data processing. Our first system adopts LAD, a delexicalisation technique that relies on multilingual embeddings for delexicalisation and uses post-editing to adapt the values to the text during relexicalisation. Our second system instead employs a SentencePiece model to perform word segmentation. In addition, we submitted a third system that combines the previous two by selecting the *LAD* output when the input contains an unseen property or entity, and the *SP* output otherwise.

The shared task's automatic evaluation shows that overall *LAD* perform better than *SP* in English as it is more robust in dealing with unseen cases. However, this is not demonstrated in the Russian results as the test set contains only instances that have been seen during training.

References

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.

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.

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.

Thiago Castro-Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussalem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional webnlg+ shared task: Overview and evaluation results (webnlg+ 2020). In *Proceedings of the 3rd WebNLG Workshop on Natural Language Generation from the Semantic Web (WebNLG+ 2020)*, 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*.

Emilie Colin and Claire Gardent. 2019. Generating text from anonymised structures. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 112–117, Tokyo, Japan. Association for Computational Linguistics.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A comprehensive survey of multilingual neural machine translation. *arXiv preprint arXiv:2001.01115*.

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.

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.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. 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.

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.

Anastasia Shimorina and Claire Gardent. 2019. Surface realisation using full delexicalisation. 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 3086–3096, Hong Kong, China. Association for Computational Linguistics.

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.

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.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*.