# Machine Translation Aided Bilingual Data-to-Text Generation and Semantic Parsing

**Oshin Agarwal***
University of Pennsylvania
oagarwal@seas.upenn.edu

**Mihir Kale**
Google
mihirkale@google.com

**Heming Ge**
Google Research
hemingge@google.com

**Siamak Shakeri**
Google Research
siamaks@google.com

**Rami Al-Rfou**
Google Research
rmyeid@google.com

## Abstract

We present a system for bilingual Data-To-Text Generation and Semantic Parsing. We use a text-to-text generator to learn a single model that works for both languages on each of the tasks. The model is aided by machine translation during both pre-training and fine-tuning. We evaluate the system on WebNLG 2020 data[1], which consists of RDF triples in English and natural language sentences in English and Russian for both the tasks. We achieve considerable gains over monolingual models, especially on unseen relations and Russian.

## 1 Introduction

Text corpora and structured data are important to practical NLP applications. They complement each other in content and topic coverage, therefore, it is important to convert from one to the other depending on the downstream task.

RDF is a common format used to store structured data as triples of `(subject, relation, object)`. Many template based, pipeline based, statistical and neural systems have been proposed for converting triples into natural text (van der Lee et al., 2018; Castro Ferreira et al., 2019; Shimorina and Gardent, 2018). Recently, an end-to-end text generation model T5 was shown to achieve state-of-the-art performance on this task (Kale, 2020). T5 has an encoder decoder framework pretrained on web pages by masking spans of text randomly and finetuned on task specific data. Several systems have also been proposed for the reverse task of semantic parsing that involves extracting RDF triples from natural text (Kamath and Das, 2018), however T5 has not been utilized for it yet.

We use T5 as the initial model, augmenting both pretraining and finetuning with several parallel corpora in a multi-task setup. This multi-task multilingual (pre-)training regimen improves the quality of both text generation and semantic parsing.

In this paper, we present the description of our multilingual multi-task system and its performance on the WebNLG 2020 challenge. This dataset consists of RDF triples in English and natural language sentences in English and Russian. We demonstrate that multitask learning across languages provides significant improvement on both the tasks, especially on unseen relations and Russian. Furthermore, incorporating parallel corpora improves the quality of text generation.

## 2 Data

### 2.1 WebNLG

WebNLG dataset consists of two different tasks:

- **Text generation**: generating a sentence that include all information in a set of triples.
- **Semantic parsing**: extracting one or more RDF triples from a natural sentence.

The first version of the dataset, WebNLG 2017 (Gardent et al., 2017), included only the task of text generation from triples to sentences in English. The second iteration, WebNLG 2020 (Castro-Ferreira et al., 2020) includes both tasks where the triples are in English and natural sentences are available in two languages: {Russian, English}.

The dataset has several auxiliary attributes; to facilitate the discussion, we split them as follows:

**WebNLG Main Corpus (WMC)** This corpus consists of a set of RDF triples and multiple corresponding sentences/references in natural text. Each reference is used as a separate instance during training. For evaluation, a set of triples is scored against each of its references and the best one is selected, as is standard with multi-reference evaluation. WMC

*Work done during internship at Google

[1] https://webnlg-challenge.loria.fr/challenge_2020/

*3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+), Dublin, Ireland (Virtual), 18 December 2020, pages 125–130,* 

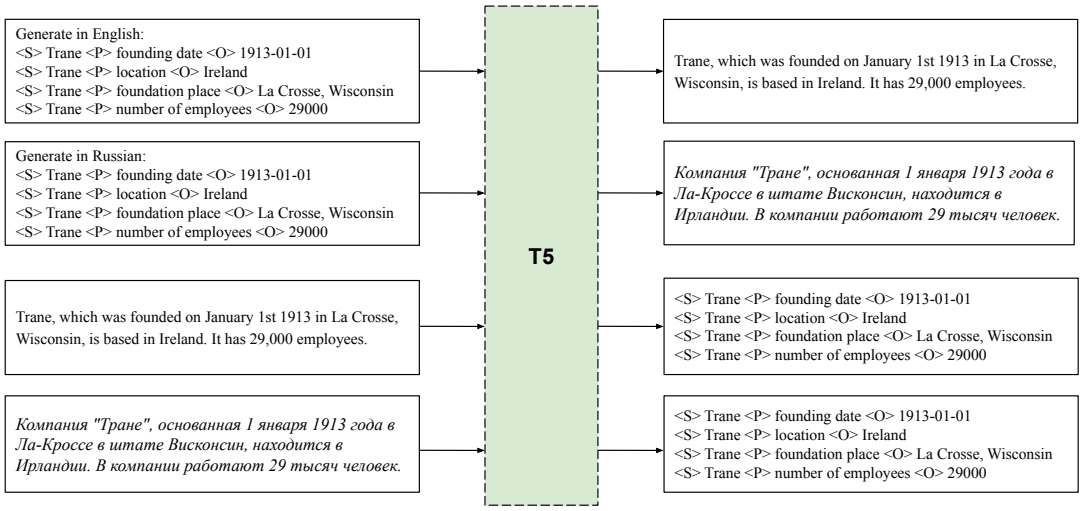


Figure 1: T5 for triples to text generation and semantic parsing using a single multi-tasked model for English and Russian. Each box represents a contiguous piece of text and triples have been shown in separate lines only for visualization.

consists of two subcorpora: i) **en-corpus**: (English triples, English sentence), and ii) **ru-corpus**: (English triples, Russian sentence).

**WebNLG Parallel Corpus (WPC)** This corpus consists of parallel sentences and entities in English and Russian. It is extracted from the Russian part of WMC. Every Russian sentence/entity is aligned with a corresponding English sentence/entity.

### 2.2 Parallel Corpus

**WMT-News corpus (WMT)** This is a parallel corpus of News data in 18 languages derived from the OPUS corpus (Tiedemann, 2012) for WMT 2019 (Barrault et al., 2019). We use the English-Russian part of the corpus.

## 3 Experimental Setup

### 3.1 Triple processing

To simplify the learning process, we pre-process the triples before feeding them to the model. First, we convert all relations from camelCase formatting to multi-word expressions (e.g. foundationPlace → foundation place). Second, we break all multi-word expressions that are joined by an underscore (e.g. La_Crosse,_Wisconsin → La Crosse, Wisconsin). Third, we serialize each triple as `<S> Subject <P> Relation <O> Object.` Finally, we concatenate all the triples in an instance as a single string. For example, Figure 1 shows four triples that are first processed and then concatenated together in a single string as input to the model. They are shown in different lines for easy visualization.

In semantic parsing, the generated triples will be in the new serialization format. Therefore, we post-process the model output and reverse the changes we introduced in preprocessing.

| | English | Russian |
|---|---|---|
| seen dev | 1669 | 793 |
| unseen dev | 455 | 1395 |

Table 1: Number of examples in the validation sets

### 3.2 Unseen Relations

The validation split in the dataset does not include unseen relations. This shortcoming will limit our ability to measure the model capacity to generalize to new relations. Therefore, we split the training set into two parts to create an unseen validation set. We reserve all examples from the training set with relations that are not seen in the original validation set to be a part of the unseen validation set. If an example has multiple triples, it is considered unseen if it has at least one unseen relation. We refer to the original validation set as *seen dev* and the new reserved validation set as *unseen dev*. The remaining examples in the training set constitute the training split for all the experiments and are used for finetuning. Table 1 shows the distribution of triples in the seen and unseen dev sets.

| Task | | Input | Output | Weight |
|---|---|---|---|---|
| triples→sentence | en-corpus | Generate in English: <en triples > | <en sentence > | 35415 |
| | ru-corpus | Generate in Russian: <en triples > | <ru sentence > | 14612 |
| sentence translation | WPC en→ru | Translate to Russian: <en sentence> | <ru sentence> | 14612 |
| | WPC ru→en | Translate to English: <ru sentence> | <en sentence> | 14612 |
| entity translation | WPC en→ru | Translate to Russian: <en entity> | <ru entity> | 637 |
| | WPC ru→en | Translate to English: <ru entity> | <en entity> | 637 |

Table 2: Tasks used for finetuning along with their input/output formats and weights in the multi-task finetuning.

| Model | English | | Russian | |
|---|---|---|---|---|
| | seen | unseen | seen | unseen |
| BT5 | 67.47 | 58.48 | 46.70 | 36.60 |
| BT5+WMT | **67.61** | 59.78 | **46.69** | **41.15** |
| BT5→WMT | 67.26 | **60.02** | 45.52 | 41.14 |
| MT5 | 65.37 | 52.75 | 44.95 | 33.43 |
| MT5→WMT | 65.99 | 54.94 | 44.16 | 35.27 |

Table 3: BLEU scores on triples-to-text generation task on the dev sets with different pre-training setups.

| Model | English | | Russian | |
|---|---|---|---|---|
| | seen | unseen | seen | unseen |
| MONOLINGUAL | 66.08 | 58.42 | 40.85 | 34.31 |
| BILINGUAL | 67.01 | 58.83 | 46.65 | 39.39 |
| BILINGUAL+WPC | **67.61** | **59.78** | **46.69** | **41.15** |

Table 4: BLEU scores on the dev set for the models finetuned for text generation

## 4 Experiments

We experimented with a variety of models for both the pretraining and the finetuning stages. Here, we present the results of the different pretrained models with the best finetuned model and the different finetuned models with the best pretrained model.

### 4.1 Pretraining

We vary the pretraining setup for the T5 models and evaluate each on the text generation task (See Table 3). The experiments follow the same trend for semantic parsing and we omit their results.

1. **BT5** we pretrain T5 on both English and Russian Wikipedia for 800K steps.
2. **BT5→WMT** we train BT5 further on WMT (en,ru) parallel corpus for 100K steps.
3. **BT5+WMT** we *jointly* pretrain T5 on both datasets from step (1+2) for 800k steps.
4. **MT5** this expands BT5 to 100+ languages (Xue et al., 2020).
5. **MT5→WMT** we train MT5 further on WMT (en,ru) parallel corpus for 100K steps.

Table 3 shows that MT5 fares slightly worse than the BT5 models. The universality of MT5 comes at the expense of small but noticeable quality degradation of generation. Moreover, we noticed that during finetuning MT5 models take longer to converge, specifically 3500 steps as opposed to 1200 steps for the BT5 model. The table also shows that adding WMT improves performance considerably on the unseen dev set, thereby producing a model with better generalization. Both BT5+WMT and BT5→WMT have very similar performance, with BT5+WMT slightly better than BT5→WMT on majority of the test splits. However, we believe that the latter setup of further pretraining on WMT is more practical than joint finetuning with WMT, especially when using mT5 as the base model. This would allow one to use the public mT5 checkpoint and continue pretraining it on WMT, which is more compute efficient than pretraining a joint model from scratch.

### 4.2 Finetuning

For the rest of the experiments we take the best pretraining set up i.e. **BT5+WMT** and finetune this model in various ways as follows:

1. **MONOLINGUAL** a model for each language.
2. **BILINGUAL** we multitask both languages and finetune a single model for both languages.
3. **BILINGUAL+WPC** We finetune the model on both languages but we add WPC for both sentence and entity translation in both directions: en → ru and ru → en.

The input and output format for each of the tasks are shown in Table 2. We add prefixes to identify the tasks, as suggested in the T5 paper. Each task is weighted by the size of its training corpus.

#### 4.2.1 Sentence Generation

The results for the generation task are shown in Table 4. While all three models have similar BLEU score on the seen dev set, we see a considerable gain in performance on the unseen dev set by multitasking the two languages as well as the sentence and entity translation in WPC.

| Text Generation | |
|---|---|
| (Acura_TLX, manufacturer, Honda), (Acura_TLX, engine, Honda_J_engine) | Honda is the manufacturer of the Acura TLX which has a Honda J engine. |
| (Acharya_Institute_of_Technology, director, Dr._G._P._Prabhukumar), (Acharya_Institute_of_Technology, campus, In_Soldevanahalli,_Acharya_Dr._Sarvapalli_Radhakrishnan_Road,_Hessarghatta_Main_Road,_Bangalore) | The Director of the Acharya Institute of Technology is Dr G P Prabhukumar and the campus is located at In Soldevanahalli, Acharya Dr. Sarvapalli Radhakrishnan Road, Hessarghatta Main Road, Bangalore. |
| **Semantic Parsing** | |
| Finland is home to the Finns and the icebreaker Aleksey Chirikov, built at the Arctech Helsinki Shipyard. | (Aleksey_Chirikov_(icebreaker), builder, Finland), (Finland, demonym, Finn), (Aleksey_Chirikov_(icebreaker), builder, Arctech_Helsinki_Shipyard) |
| Пенджаб, Пакистан, возглавляется Провинциальной ассамблеей Пенджаба. [English translation: Punjab, Pakistan is led by the Provincial Assembly of Punjab.] | (Punjab,_Pakistan, leaderTitle, Provincial_Assembly_of_the_Punjab) |

Table 5: Examples of text generation and semantic parsing by respective final models.

| Model | English | | Russian | |
|---|---|---|---|---|
| | seen | unseen | seen | unseen |
| MONOLINGUAL | 53.54 | 27.86 | 53.68 | 15.78 |
| BILINGUAL | **59.49** | **31.44** | 56.12 | **25.87** |
| BILINGUAL+WPC | 52.56 | 31.23 | **56.15** | 25.75 |

Table 6: Full Triple match F1 on the dev set for the models finetuned for semnantic parsing

| Model | English | | Russian | |
|---|---|---|---|---|
| | seen | unseen | seen | unseen |
| MONOLINGUAL | 89.50 | 77.26 | 88.60 | 62.94 |
| BILINGUAL | **94.96** | **79.96** | **91.40** | 82.32 |
| BILINGUAL+WPC | 88.79 | 79.63 | 90.64 | **82.84** |

Table 7: Triple elements strict match F1 on the dev set for the models finetuned for semantic parsing

We did not observe any common failure cases. Most errors seemed to be a different way to express the same triples correctly, not captured in the references, or the lack of enough fluency in some cases. Some examples are shown in Table 5.

#### 4.2.2 Semantic Parsing

The system generates the serialized triples in the same format that was input to the RDF verbalizer, which are then post-processed. Evaluation[2] is done at the triple level using full triple match F1 (Table 6) and at an element level using strict matching of the exact string and the element type (Table 7). The BILINGUAL system performs better than the MONOLINGUAL ones but adding WPC leads to a drop in performance on the English seen set and has no impact on the other dev splits. Some examples of extracted triples are shown in Table 5.

We observe a common failure pattern, where the model extracts the correct triple but interchanges the subject and object. For example, the model extracts (New York City, isPartOf, Manhattan) instead of (Manhattan, isPartOf, New York City). We hypothesize that this is due to subjects and objects often appearing in a specific order in the training sentences. However, they cannot be shuffled since these are natural sentences. One could potentially generate paraphrases, especially changing the voice of the sentence, to augment the training data to improve on this. Another challenge was postprocessing to get the canonical name. Some target entities in the dataset are in quotes with spaces between multiple words instead of underscores, unlike the rest of the dataset and our postprocessing is unable to capture these in the expected format. However, such differences in canonical name formats seem to be small in number.

#### 4.2.3 Multitasking Generation and Parsing

We tried a multi-task setup with both Data-To-Text Generation and Semantic Parsing tasks but this led to a drop of ~5 BLEU on both seen sets and ~2 BLEU on English unseen.

## 5 Final Results

In prior sections, all the experiments were evaluated on the seen and reserved unseen dev sets. In this section, we train the model on the full training set and report the results on the test sets, as communicated by the challenge organizers. We use our best pre-trained model BT5+WMT and our best finetuned model BILINGUAL+WPC for text generation and BILINGUAL for semantic parsing. The results are shown in Table 8. For English, results are broken into seen and unseen categories or sub-domains. Unseen categories might have relations seen in a different category, unlike the unseen

[2] https://github.com/WebNLG/WebNLG-Text-to-triples

| Metric | English seen | English unseen | English total | Russian total |
|---|---|---|---|---|
| **Text Generation** | | | | |
| BLEU | 61.08 | 43.98 | 51.74 | 51.63 |
| METEOR | 43.30 | 39.30 | 41.11 | 67.60 |
| chrf++ | 72.50 | 63.60 | 67.90 | 68.30 |
| TER | 39.10 | 47.00 | 43.50 | 42.00 |
| BERTScore F1 | 96.30 | 94.70 | 95.40 | 90.70 |
| BLEURT | 60.00 | 56.00 | 60.00 | - |
| **Semantic Parsing** | | | | |
| Strict | 87.66 | 53.87 | 67.55 | 91.09 |
| Exact | 87.70 | 55.06 | 68.19 | 91.11 |
| Partial | 88.30 | 60.93 | 71.35 | 91.71 |
| Entity-type | 88.75 | 65.32 | 73.71 | 92.31 |

Table 8: Evaluation on the test set, broken by seen and unseen categories. Pretraining uses BT5+WMT. Finetuning uses BILINGUAL+WPC for text generation and BILINGUAL for semantic parsing.

dev set we created with unseen relations across all categories.

**Automatic Evaluation** We report BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), TER (Snover et al., 2006), chrF++ (Popović, 2017), BERTScore (Zhang et al., 2019) and BLEURT (Sellam et al., 2020) for text generation, and the element level strict, exact, partial and entity type match F1 for semantic parsing[3], using the evaluation platform in (Moussalem et al., 2020). In both cases, results are as expected and follow the same trend as the dev set.

**Human evaluation** The organizers performed a human evaluation on the system output for the text generation tasks. Annotators rated the generated text in each example on five aspects on a scale of 0 to 100. Following are the different aspects –

1. Coverage: Generated text covers all the relations in the triples.
2. Relevance: Generated text covers only the relations in the triples.
3. Correctness: Generated text has the correct subjects and objects for the relations.
4. Structure: Text is grammatical and well-structured.
5. Fluency: Text is fluent and sounds natural.

Annotator scores were averaged and are shown in Table 9. The organizers also normalized the scores across the submitted systems and clustered

[3] https://webnlg-challenge.loria.fr/challenge_2020/#automatic-evaluation

| Metric | English seen | English unseen | English total | Russian total |
|---|---|---|---|---|
| **Text Generation** | | | | |
| Coverage | 89.31 | 85.22 | 86.69 | 95.42 |
| Relevance | 89.06 | 87.64 | 88.18 | 94.42 |
| Correctness | 88.77 | 85.25 | 86.34 | 95.48 |
| Structure | 87.17 | 84.76 | 85.64 | 95.62 |
| Fluency | 83.28 | 81.05 | 82.30 | 93.13 |

Table 9: Human evaluation on text generation. Scores are on a scale of 0-100. Pretraining uses BT5+WMT and Finetuning uses BILINGUAL+WPC .

systems such that there was no statistically significant differences within a cluster according to the Wilcoxon rank-sum significant test. On English, we ranked 1 on relevance, correctness and structure, and ranked 2 on coverage and fluency. On Russian, we ranked 1 on all five aspects.

## 6 Conclusion

We developed a system for bilingual data-to-text generation and semantic parsing, using T5, aided by machine translation during both pretraining and finetuning. We evaluated this system on WebNLG 2020 and showed improvements, especially on unseen relations.

## Acknowledgments

We thank Scott Roy, Yun-Hsuan Sung and the anonymous reviewers for their valuable feedback.

## References

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 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.

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

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.

Mihir Kale. 2020. Text-to-text pre-training for data-to-text tasks. *arXiv preprint arXiv:2005.10433*.

Aishwarya Kamath and Rajarshi Das. 2018. A survey on semantic parsing. *arXiv preprint arXiv:1812.00978*.

Chris van der Lee, Emiel Krahmer, and Sander Wubben. 2018. Automated learning of templates for data-to-text generation: comparing rule-based, statistical and neural methods. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 35–45, Tilburg University, The Netherlands. Association for Computational Linguistics.

Diego Moussalem, Paramjot Kaur, Thiago Castro-Ferreira, Chris van der Lee, Conrads Felix Shimorina, Anastasia, Michael Röder, René Speck, Claire Gardent, Simon Mille, Nikolai Ilinykh, and Axel-Cyrille Ngonga Ngomo. 2020. A general benchmarking framework for text generation. In *Proceedings of the 3rd WebNLG Workshop on Natural Language Generation from the Semantic Web (WebNLG+ 2020)*, Dublin, Ireland (Virtual). 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. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

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.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.