Semantic Triples Verbalization with Generative Pre-Training Model

Pavel Blinov
Sber Artificial Intelligence Laboratory / Moscow, Russia
Blinov.P.D@sberbank.ru

Abstract

The paper devoted to the problem of automatic text generation from RDF triples. This problem was formalized and proposed as a part of the 2020 WebNLG challenge. We describe our approach to the RDF-to-text generation task based on a neural network model with the Generative Pre-Training (GPT-2) architecture. In particular, we outline a way of base GPT-2 model conversion to a model with language and classification heads and discuss the text generation methods. To research the parameters' influence on the end-task performance a series of experiments was carried out. We report the result metrics and conclude with possible improvement directions.

1 Introduction

The idea of semantic web has a long history of research and development. The Resource Description Framework (RDF) is the most common and known standard for semantic data interchange developed by the World Wide Web Consortium (Lassila et al., 1998). The RDF data model allows to encode knowledge in a form of *(subject, predicate, object)* statements known as triples. Thus providing a way of creating common knowledge databases understandable for machine and human.

This idea together with recent development of transformer language models unlock a set of promising research directions. Two of them were highlighted at this year's workshop on Natural Language Generation from the semantic Web (WebNLG) (Castro Ferreira et al., 2020). The WebNLG organizers proposed: 1) RDF-to-text generation task and 2) Text-to-RDF semantic parsing task (reverse of the first one). Each task was suggested for English and Russian language.

Basically the first task is the following: given an input RDF triples set, build a system to yield its verbal logical equivalent in natural language. For example, given the RDF set of three triples:

Adare Manor|completionDate|1862
Adare Manor|architect|Augustus Pugin
Adare Manor|buildingStartDate|1700

a good answer would be *"The construction of Adare Manor began in 1700 and was completed in 1862. The manor was designed by Augustus Pugin."*

Such a system, if proven to show human comparable performance, potentially can have several practical applications. Based on the system a tool for specific domain RDF databases verbalization can be created. Generally making interaction with knowledge graphs and databases more natural for lay users. Another use case includes an application in the dialog systems development process. Often conversational agents use retrieval-based (Ji et al., 2014; Yang et al., 2018) or hybrid (Yang et al., 2019) approaches for best utterance selection. Those approaches require a predefined candidate response set, which can be automatically verbalized from an appropriate RDF database.

In this work, we restrict ourselves only to the RDF-to-text generation task for the Russian language. Our ultimate goal was to benchmark performance of a neural network language model with GPT-2 architecture (Radford et al., 2019) together with some decoding methods.

2 Data

The workshop organizers provided the dataset (in the XML file format) for each language along with the split: *train*, *dev* and *test*. The Russian data was collected from 9 DBpedia (Auer et al., 2007) categories. Absolute numbers for each category and split part can be found in Table 1.

Basically each data entry may contain up to 7 RDF triplets and a few variants of this triplet set

3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+), Dublin, Ireland (Virtual), 18 December 2020, pages 154–158,

Category	train	dev	test
Airport	949	136	190
Astronaut	463	66	92
Building	852	120	167
CelestialBody	555	79	109
ComicsCharacter	250	35	50
Food	1,231	175	245
Monument	234	30	44
SportsTeam	684	98	134
University	355	51	71
Total	5,573	790	1,102

Table 1: Statistic details of the Russian dataset.

verbalization. Based on the *train* data, the median value of an entry triplets is 3. Except for a few outliers the number of verbalisations for an entry also no more than 3. Each entry also includes some additional parameters, for example specs on triplet positional arrangement, but we didn't use it except for the translation links. To stay in the single language space we also had to explicitly translate predicate statements (227 unique elements) thus finalizing triplets conversion to the Russian language.

Furthermore, as there were no restrictions on external data use, we tried to experiment with additional data (see Section 3.2). For this purpose Baidu SKE dataset[1] was automatically translated from Chinese to Russian[2] and used as an augmentation data. The dataset (much larger is size) has a similar structure, each entry is a set of associated tuples *(subject, predicate, object)* and a text description. The number of unique predicate statements is 50 and there are 194,747 entries in total.

3 Proposed approach

Since their introduction the neural network models with transformer-based architecture (Vaswani et al., 2017) has become the default choice when approaching a natural language processing problem. This success primarily is the performance boost and usability across a wide range of tasks. The same base model can be finetuned for a specific task (in an end-to-end fashion) with a low cost effort and relatively small amount of data.

In this work we exploit the same strategy by taking Russian GPT-2 model[3] (24 transformer block layers, 1024-dimensional word embeddings and 16 self-attention heads, with about 350M parameters in total) as base and finetune it for the RDF-to-text generation task.

The GPT-2 essentially is a language model that is trained to predict the next token (a sub-word) given all previous tokens of a sequence. Technically the cross-entropy loss function over n vocabulary tokens $\{t_1, ..., t_n\}$ used to estimating the conditional probability distribution $p(t_i|t_1, ..., t_{i-1})$. But we cannot directly apply such a model to our task as the genuine GPT-2 model has a lack of control over the generation process and succeeds only in spawning of text endings given a beginning. Our goal is to force the model to wrap (join and rephrase) the given RDF set triplets in the form of coherent, syntactically and semantically correct natural language phrases. For this purpose we defined the model with double heads. First one is the same (as in GPT-2) language head and the second one is the classification head. The aim for the second head is exactly to distinguish between correct or wrong (a sample was randomly selected to represent a negative example) verbalisation of an input RDF set. Thus our final loss function is the sum of two cross-entropy components. By minimizing such loss our model learns to automatically translate an RDF set to an adequate natural language phrase.

The sample input data for the model is a pool of indices lists and masks. The main part of this pool is the list of token indices of an input sequence, which could be broken down into two parts: RDF-part and phrase-part. These parts are separated from each other by the special mediator token (by start/end tokens analogy). In the inference mode there is only an RDF-part and the mediator token is the trigger to start a phrase generation process (Section 3.1). We trained the unified model for all categories and triplet cardinalities.

3.1 Text generation stage

Once the model has been trained it can generate verbalisations for unseen RDF triplet sets. Basically it is an iterative process of tokens' probability estimation and the "best" token selection, until the stop criteria is reached (e.g. the terminate token is encountered). The obvious strategy is greedily choosing the most probable token at each step. But it has major drawbacks of text repetitions and obvious phrases selection. Instead of a greedy al-

[1] http://ai.baidu.com/broad/introduction

[2] https://yadi.sk/d/P55m92dEyC3w8g

[3] In shortage of pre-trained GPT-2 models for the Russian language we had to train one from scratch.

gorithm, we use a beam search algorithm which implements the idea of keeping and scoring an alternative set of beams during the generation process. Thus allowing immediately un-obvious variants to evolve further to high quality ones.

As recently highlighted in Holtzman et al. (2020) the high quality natural language text has an irregular structure and is much less predictable in terms of simple token probability maximization strategy. To introduce more variability and improve the quality of generated text particular sampling methods (*Top-K* and *Top-P* sampling) have been proposed (Fan et al., 2018; Holtzman et al., 2020). Their overall idea is to choose a random token according to the probability distribution from only a fixed top-range of high probable candidates. The *Top-K* parameter explicitly limits the range to *K* tokens. And the *Top-P* parameter defines the tokens range by cumulative probability value *P*. Also the overall probability distribution shape can be controlled with the *Temperature* parameter (Ackley et al., 1985).

The above mentioned methods do not contradict with each other and could be jointly implemented in a single text generation pipeline. Due to the sampling methods this pipeline can be viewed as a randomized process. For an input RDF set the pipeline can be executed *Run-K* times and each run samples a verbalisation candidate. We've done some experiments with this *Run-K* and other generation parameters (Section 3.2). By analogy with the beam search algorithm we can estimate the candidate's score (as the probability product of its tokens). And the final RDF's verbalisation was selected by this score maximization.

Finally, we've applied some post-processing (with regular expressions) on the final candidates to overcome tokenization artefacts, like paired characters concatenation, a float delimiter unification, etc.

3.2 Experimental results

The evaluation of systems that produce natural language text as a result is a non-trivial problem. Several metrics have been designed to quantitatively assess such systems' performance. For this challenge the organizers provided the evaluation tool with five base metrics (Castro Ferreira et al., 2020). In this study the BLEU metric (Papineni et al., 2002) was selected as the primary one.

We've implemented the above mentioned model using the Transformers library (Wolf et al., 2019).

Metric	dev	test	synthetic
BLEU	35.7	43.1	38.5 ± 0.8
BLEU NLTK	35.4	43.0	38.9 ± 0.8
METEOR	51.6	57.6	60.0 ± 0.5
chrF++	54.3	59.5	61.8 ± 0.4
TER	55.1	48.7	54.4 ± 0.6
BERT-score P	87.8	89.8	87.7 ± 0.1
BERT-score R	85.0	87.3	88.0 ± 0.1
BERT-score F_1	86.2	88.4	87.9 ± 0.1

Table 2: Systems performance metrics (%).

And trained it (from the base GPT-2 model) with the Adam optimizer (initial learning rate 3×10^{-5}), an input sequence length of 512 tokens and the batch size of 2 examples for 3 epochs on a Tesla V100 GPU. Alternatively we've tried to first make a pre-training on the translated RDF data (see Section 2) and then finish the training with this challenge dataset. But this strategy did not seem to affect the performance much (see the dashed line on the *a*-part of Figure 1) so we abandoned this approach.

In our experiments the generation process was governed by four main parameters: *Run-K*, *Temperature*, *Top-K* and *Top-P*. The Figure 1 depicts dependency between these parameters values (x-axis) and the BLEU metric (y-axis) on the *dev* set. In each case only a single parameter value was varied while the other ones were fixed. As can be seen from the Figure 1 the performance correlations are not so explicit, especially for the *Temperature* parameter values. Our best guess is that the RDF-to-text task is much simpler than an open-ended language generation and these parameters' values are less important. The number of pipeline runs *Run-K* parameter seems the most influential one, but also values after 11 practically yield the same BLEU. For the final generation run on the *test* data we used the following parameters: *Run-K=19*, *Temperature=0.7*, *Top-K=11*, *Top-P=0.9*. Full list of metrics for the values of generation parameters are shown in Table 2 (*dev* and *test* columns).

In an attempt to get an idea about performance range values in this task we made the following experiment. As the majority of entries contain several verbalisation variants the one can be randomly picked as a true variant and the other would be references. We have joined *train* and *dev* data in the single pool and randomly sampled a batch of 790 entries (the size of *dev* set) from it, followed by

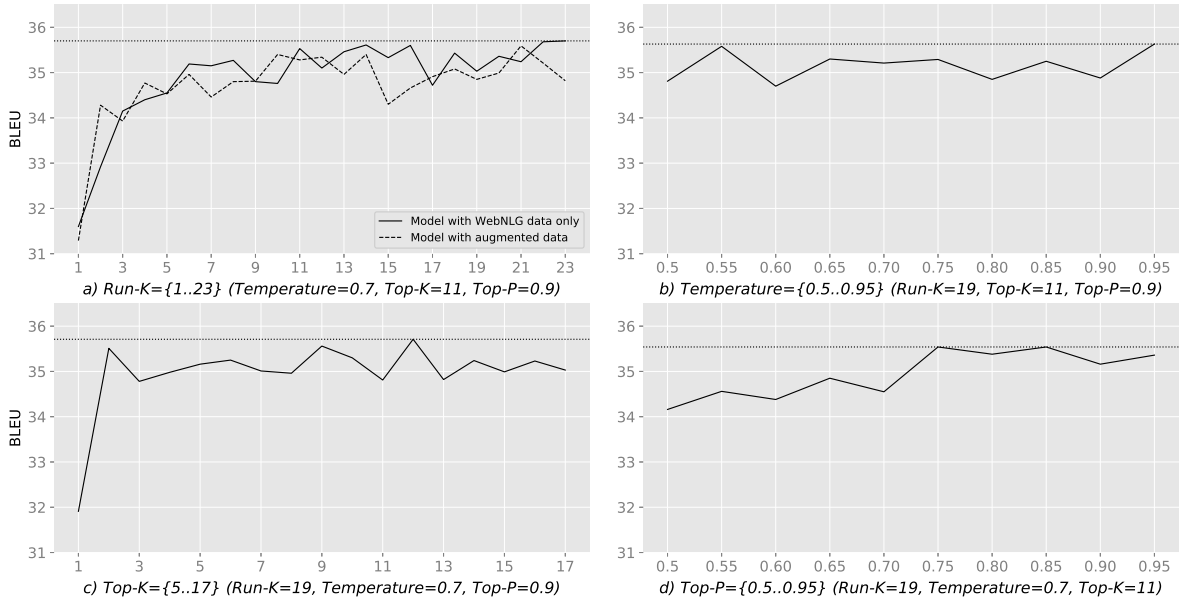

Figure 1: Model performance (BLEU-metric, %) on the validation dataset.

the metrics computation. The sampling and evaluation procedures were repeated 37 times. From the results we can compute average metrics along with standard deviations. These values are shown in column *synthetic* of the Table 2. The comparison with *dev* metrics shows that the proposed model has a potential room for improvement.

Further, the large gap between *dev* and *test* metrics implies that entries' distribution substantially differs. We suppose the shift was caused by the way of *test* data selection, because only RDF triples with the entities and categories already seen in the training data was included in the *test*.

Beside automatic evaluation metrics the organizers performed the human evaluation of participants results according to five criteria: *data coverage*, *relevance*, *correctness*, *text structure* and *fluency* (Castro Ferreira et al., 2020). Furthermore the baseline system (Moussallem et al., 2020) also was assessed in terms of this criterias. Our system was able to overcome the baseline for all criterias except the *data coverage*.

4 Conclusions

In this paper we presented our RDF triples verbalization system based on the neural network model with GPT-2 architecture. Specifically the model with language and classification heads were used to generate appropriate text given a set of Russian triples. We described the model implementation details and briefly discussed common decoding methods and its recent sampling variants.

The data provided in WebNLG challenge allowed us to train the model and experiment with the generation pipeline. From this we can conclude that the verbalisation task from RDF data is more restricted (compared to open-ended language generation) and sustainable results can be obtained from a wide range of generation parameters. At the same time metric comparison with the other participant's systems and synthetic benchmark revealed the limitation of such approach.

While this model can be used for verbalisation candidate generation the procedure of best candidate selection obviously should be further improved. A recap of processed test entries showed that sometimes not all parts of triples are mentioned in text or repetitions are encountered, while the candidate pool contained more appropriate variants. The mediocre results of human evaluation on the *data coverage* criteria are another confirmation of such system inefficiency. We'll consider to build a separate ranking model which hopefully would be able to better distinguish such cases.

References

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cogn. Sci.*, 9(1):147–169.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data.

In *The Semantic Web*, pages 722–735, Berlin, Heidelberg. Springer Berlin Heidelberg.

Thiago Castro Ferreira, Claire Gardent, Chris van der Lee, Nikolai Ilinykh, 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.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *CoRR*, abs/1408.6988.

Ora Lassila, Ralph R. Swick, World Wide, and Web Consortium. 1998. Resource description framework (rdf) model and syntax specification.

Diego Moussallem, Paramjot Kaur, Thiago Castro Ferreira, Chris van der Lee, Anastasia Shimorina, Felix Conrads, 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 on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *Computing Research Repository*, arXiv:1910.03771.

Liu Yang, Junjie Hu, Minghui Qiu, Chen Qu, Jianfeng Gao, W. Bruce Croft, Xiaodong Liu, Yelong Shen, and Jingjing Liu. 2019. A hybrid retrieval-generation neural conversation model. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM'19, page 1341–1350, New York, NY, USA. Association for Computing Machinery.

Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*.