The UPC RDF-to-Text System at WebNLG Challenge 2020

David Bergés, Roser Cantenys, Roger Creus, Oriol Domingo, José A. R. Fonollosa
Universitat Politècnica de Catalunya, Barcelona
TALP Research Center
{david.berges, roser.cantenys
roger.creus, oriol.domingo.roig}@estudiantat.upc.edu
{jose.fonollosa}@upc.edu

Abstract

This work describes the end-to-end system architecture presented at WebNLG Challenge 2020. The system follows the traditional Machine Translation (MT) pipeline, based on the Transformer model, applied in most text-to-text problems. Our solution is enriched by means of a Back Translation step over the original corpus. Thus, the system directly relies on lexicalise format since the synthetic data limits the use of delexicalisation.

1 Introduction

Natural Language Generation (NLG) can be divided into: text-to-text generation or data-to-text generation, according to Gatt and Krahmer (2017). The WebNLG Challenge 2020 consists in mapping data-to-text. More specifically, the data is a set of Resource Description Framework (RDF) triples extracted from DBpedia and the corresponding text is a verbalisation of these triples [1].

The information structure is based on RDF, which consist of three elements: ⟨*subject, predicate, object*⟩. Thus, it establishes relations (*predicate*) between entities (*subject, object*). This can be appreciated in Figure 1. However, this information structure is not easy readable neither understandable, hence, it is hard for people to comprehend the meaning of such data.

There has been a lot of previous work in the NLG domain (Reiter and Dale, 2000) in the past two decades. Bontcheva et al. (2004) work in the medical domain, where they use a traditional NLG approach to generate sentences from RDF data filtering repetitive RDF, and then group coherent triples aggregating the generated sentences in order to produce the final ones. (Cimiano et al., 2013) generate cooking recipes from semantic web data, using a large corpus to extract lexicon in the cooking domain, which is then used in conjunction with a traditional NLG approach to generate cooking receipts. (Duma and Klein, 2013) use a method which works well on RDF triples in a seen domain but fails with unseen ones. Their aim is to learn sentence templates from parallel RDF data and text corpora by means of aligning entities in RDF triples with entities mentioned in sentences, and then extracting these templates from the aligned sentences by replacing the entity men- tion with a unique token.

We decided to only participate in the English version of the RDF-to-Text challenge (Castro Ferreira et al., 2020). We used a model based on the Transformer encoded-decoder architecture (Vaswani et al., 2017). Moreover, inspired by previous work in the MT field, we enlarged the original corpus by means of Back Translation (BT) (Sennrich et al., 2016).

The rest of the document is organised as follows. First, in Section 2 we take a deeper dive into the task formulation. Next, in Section 3 the preprocessing plan is explained. Then, in Section 4 we depict the Transformer model architecture adapted to our problem. Thereafter, we briefly describe postprocessing in Section 5. Finally, in Section 6 we summarize the implementation of BT over the original challenge followed by brief results and conclusions in Sections 7 and 8 respectively.

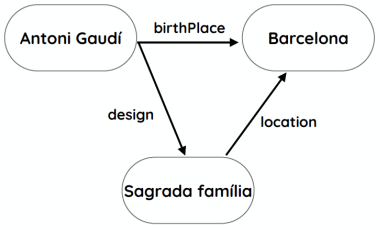
2 Task Formulation

The goal of the RDF-to-Text task is to generate text from a set of triples, which are words establishing relations between them.

The input to our system is data in the form of triples that can be denoted as a set of RDF, i.e. $\mathcal{K} := \{r_1, ..., r_n\}$. Each RDF r_i can be defined as $\langle s_i, p_i, o_i \rangle$, these elements stand for subject, pred-

[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 167–170,



(a) Knowledge graph.

RDF Triples	<Antoni Gaudí, design, Sagrada Família> <Antoni Gaudí, birthPlace, Barcelona> <Sagrada Família, location, Barcelona>

(b) Knowledge base and its RDF triples.

Natural Sentence	Antoni Gaudí was born in Barcelona, the city where he designed Sagrada Família.

(c) Corresponding natural language description.

Figure 1: Example of a knowledge graph (a) with its corresponding RDF triples (b) and its natural language description (c).

icate and object, respectively. Notice that each element can contain more than one word, there is no prior restriction in that sense. For instance, the subject 'Barack Obama' would be encoded as $s_i = [Barack, Obama] = s_{ij} = [s_{i1}, s_{i2}]$, so i indicates the RDF in $\mathcal{K}$ and index j denotes the word position in each subject, predicate or object element.

Finally, we aim to generate a discourse $\mathcal{S}$, which consists of a sequence of words $[w_1, ..., w_m]$. The resulting discourse in $\mathcal{S}$ should be grammatically correct and should also contain all the information present in the triples.

3 Preprocessing

In this section we describe the first steps performed on data. There is a common step, delexicalisation, that has been performed since the very beginning of this challenge, back to 2017 [2]. We decided to avoid this step due to the implementation of our BT method that does not contain the required mapping: from individual entities to generic words.

The very first data processing guide is defined as follows, and it is exemplified in Table 1. First of all, we linearise the RDF input and split the camelCase notation. Then, Moses Tokenizer (Koehn et al., 2007) is applied to separate punctuation from words, preserving special tokens such as dates, and normalize characters. Finally, Byte Pair Encoding (BPE) (Sennrich et al., 2015) is applied to enable the model to be more robust to unseen data. This is a traditional technique that increases the translation quality of models. BPE is learned with the training plus validation procedure and is used for the source and target vocabulary. This way, the model is trained for both receiving and predicting BPE encoded vocabulary, also for the test set. Finally, the system implementation allows to learn embeddings from scratch from the vocabulary that has been already encoded with BPE.

[2] https://webnlg-challenge.loria.fr/challenge_2017/

4 The Transformer Model

The Transformer (Vaswani et al., 2017) is considered a state-of-the-art encoder-decoder architecture, with great success in a vast field of applications such as MT. Following this success and taking advantage of its simpler architecture, we proposed a simple transformer approach trained in an end-to-end fashion. One of the main traits that enables these models to attain such surprising results, is the attention mechanism, that allows to model dependencies regardless their distance in the input or output sequences. This capability is a fundamental feature for RDF-to-Text, as automated generation of text takes into account the relationship between words that may not appear consecutively.

4.1 Model Parameters and Optimization

For the model's architecture, we used a total of 3 layers with 1,024-dimensional Feed Forward Networks (FFN) and 8 attention heads, performing cross + self attention at each layer. We used 256-dimensional embeddings with fixed sinusoidal positional encodings, shared across the entire network.

We used the Adam optimizer with $b1 = 0.9$, $b2 = 0.98$ and $\epsilon = 10 - 9$. We increased the learning rate linearly for a total of 4,000 warming steps to 1e-03, and decreased it following an inverse square root formula from thereon. Additionally, we applied several dropout techniques such as dropout, gradient clipping and label smoothing for our loss formula.

With this model configuration, the performed experiments concluded that the best choice was to use 7,000 subwords for the BPE encoding.

RDF Input	⟨ Baku Turkish Martyrs ' Memorial, nativeName, " Türk Şehitleri Anıtı " ⟩ ⟨ Baku Turkish Martyrs ' Memorial, location, Azerbaijan ⟩
Linearise	Baku Turkish Martyrs ' Memorial nativeName " Türk Şehitleri Anıtı " Baku Turkish Martyrs ' Memorial location Azerbaijan
camelCase Removal	Baku Turkish Martyrs ' Memorial native Name " Türk Şehitleri Anıtı " Baku Turkish Martyrs ' Memorial location Azerbaijan
BPE & Tokenization	Baku Turkish Mart@@ yrs ' Memorial native Name " T@@ ürk Ş@@ eh@@ it@@ l@@ er@@ i An@@ ı@@ t@@ ı " Baku Turkish Mart@@ yrs ' Memorial location Azerbaijan
Transformer	The Baku Turkish Mart@@ yrs ' Memorial is located in Azerbaijan . The native name of the Baku Turkish Mart@@ yrs ' " T@@ ürk Ş@@ eh@@ it@@ l@@ er@@ i An@@ ı@@ t@@ ı .
System Output	The Baku Turkish Martyrs ' Memorial is located in Azerbaijan . The native name of the Baku Turkish Martyrs ' Memorial is Turk Sehitleri Anıtı .

Table 1: Exemplification of each step in the system architecture using a test instance.

5 Postprocessing

The Transformer model outputs a sequence of predicted words, then, the system removes the tokenization as well as BPE. One example of the output of the system is shown in Table 1.

6 Back Translation

BT runs in a semi-supervised environment where both parallel corpora and monolingual data in the target language are available (Sennrich et al., 2015). First, BT trains an intermediate system on the parallel data which is used to translate the target monolingual data into the source language, i.e. text-to-RDF. The latter, results in a parallel corpus where the source is synthetic MT output while the target is genuine text written by humans. Afterwards, the generated synthetic parallel corpus is added to the real bitext in order to train a final model that will translate from the source to the target language, equivalently RDF-to-text.

The parallel dataset was already provided by the challenge and contains the translation from RDF-to-text and vice-versa. Hence, we just needed an external monolingual corpus of the target language to perform augmentation of the source data. In order to do so, we implemented a distance-based approach to the training data since entities appearing in the corpus were annotated. Taking this into account, not only was the team capable of scrapping Wikipedia pages of most similar entities to the ones in the original corpus, but we were also able to limit scrapping to the first three paragraph in each page without loss of quality. In order to determine the most similar entities, an embedding distance was computed regarding Wikipedia2Vec (Yamada et al., 2020) that allows to query for entities rather than words.

The current approach to solve the Back-Translation, text-to-RDF, implies using parsing trees that guarantee that elements in the RDF appear in the text. Consequently, this implementation generated coherent data with respect to text. The final dataset, which integrated the real corpus and the synthetic one, has around 340,000 instances.

A detailed description of this experiment can be found in (Domingo et al., 2020).

7 Results

In Table 2, we show the results obtained in the test set for the competition. One remarkable aspect is that there is not a significant difference between the performance in the seen and unseen domain regarding the METEOR, TER and chrF++ metric. On the other hand, there exists a performance drop based on BLEU score in the unseen data with respect to the seen one.

Data	**BLEU** (↑)	**METEOR** (↑)	**chrF++** (↑)	**TER** (↓)
All	39.12	0.33	0.57	0.56
Seen Categories	51.85	0.37	0.64	0.47
Unseen Categories	29.46	0.31	0.52	0.60
Unseen Entities	35.34	0.32	0.56	0.57

Table 2: Performance of the system regarding different data partitions in the test set.

8 Open discussion

An interesting capability that was not implemented would have been to obtain delexicalised synthetic data, so the model learned more generic representations. Furthermore, it would be interesting to enlarge the synthetic corpus and use this synthetic corpus to train more relevant models in the RDF-to-Text domain.

Acknowledgment

We want to thank anonymous reviewers for their comments on the paper. This work was supported by the project ADAVOICE, PID2019-107579RB-I00 / AEI / 10.13039/501100011033.

References

Kalina Bontcheva and Yorick Wilks. 2004. Automatic report generation from ontologies: The miakt approach. In *Natural Language Processing and Information Systems*, pages 324–335, 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.

Philipp Cimiano, Janna Lüker, David Nagel, and Christina Unger. 2013. Exploiting ontology lexica for generating natural language texts from RDF data. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 10–19, Sofia, Bulgaria. Association for Computational Linguistics.

Oriol Domingo, David Bergés, Roser Cantenys, Roger Creus, and José A.R. Fonollosa. 2020. Enhancing sequence-to-sequence modelling for RDF triples to natural text. In *Proceedings of the 3rd WebNLG Workshop on Natural Language Generation from the Semantic Web (WebNLG+ 2020)*, Dublin, Ireland (Virtual). "Association for Computational Linguistics".

Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 83–94, Potsdam, Germany. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2017. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *CoRR*, abs/1703.09902.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Studies in Natural Language Processing. Cambridge University Press.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia. *arXiv preprint 1812.06280v3*.