Sequence-to-Sequence Modeling for RDF triples to Natural Text

Bergés Lladó, David
daviid.bergees@gmail.com

Cantenys Sabà, Roser
r.cantenys@gmail.com

Creus Castanyer, Roger
creus99@gmail.com

Domingo Roig, Oriol
orioldomingoroig@gmail.com

April 1, 2020

1 Description

It is known that Knowledge Data Base is one of the main form to represent structured information, as well as, ontologies. The reason why a lot of companies have recently started to use this DB format is due to the wide range of industrial applications, such as question answering systems, recommender systems, etc.

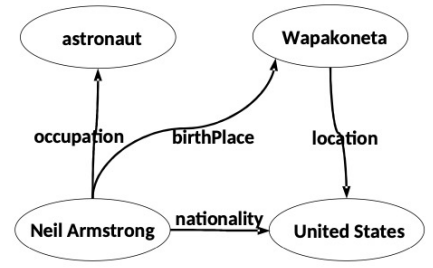
Thus, our team thinks that tackling a problem around the theme of Natural Language Generation (NLG) with Knowledge Data Base is a topic that current data scientists must be familiar with. Not only is it important for us to know how to apply Deep Learning techniques to real world problems, but also to have a first glimpse of how how complex human language generation can be.

2 Objective

The main objective of this project is to generate high-quality text from a given set of Resource Description Frameworks (RDF), i.e. triplets of the form (subject, predicate, object). A schema is provided in [Figure 2][1].

In this work, we will study the performance of a Sequence-to-Sequence model, which is know to be a baseline model for this task. This approach will help us to determine which results can be achieved by means of a simple model.

Afterwards, this baseline model will be modified in order to improve previous results. Thus, we will focus on improving benchmark results, and fortunately achieve state-of-the-art results.



RDF Triples	<Neil Armstrong, occupation, astronaut> <Neil Armstrong, nationality, United States> <Neil Armstrong, birthPlace, Wapakoneta> <Wapakoneta, Location, United States>

(a) Knowledge base and its RDF triples.

Natural Sentence	Neil Armstrong was an American astronaut born in Wapakoneta, a city in the United States

(b) Corresponding natural language description.

Figure 1: Knowledge Base with the equivalent RDF (a) and the text associated (b).

[1]Image taken from Triple-to-Text: Converting RDF Triples into High-Quality Natural Languages via Optimizing an Inverse KL Divergence.

3 Database

The following dataset contains RDF and their corresponding text, new versions have been released, however, we will make use of the original version since it has been considered the benchmark for previous scientists:

- **WebNLG**: webnlg-dataset-2017 14 April 2017, (9,674 data inputs and 25,298 data-text pairs).

We have also realised that there are some other datasets which could be used. Nevertheless, we will use this one since it is considered as the benchmark and leading models compare against it.

In addition, this database, unlike the others available, is already preprocessed so it perfectly fits our task which is focusing on natural language generation, not in data collection.

Theoretically, we will train our models from scratch, including embedding representation. Nevertheless, if we consider using pretrained embeddings for experiments or any other task, we will consider GloVe representation.

4 Initial work plan and tasks

Once the data is collected from the source mentioned above, it is recommended to preprocess data following WebNLG Challenge practices[2]. That is, turning the `xml` file into one source file, containing RDF information, and one target file, containing the corresponding text.

In this work, Facebook AI Research Sequence-to-Sequence Toolkit written in Python (Fairseq) will be used for developing. It is a software that allows researchers to train custom/referenced models for text generation tasks. Thus, our inputs and outputs files must satisfy the requirements of this software, which are Byte Pair Encoding (BPE) and Fairseq format[3].

Next step is to train referenced models provided by the Fairseq software. Thus, we will be able to conduct a study on different sequence-to-sequence models that will allow us to compare against benchmark scores, obtained in the WebNLG Challenge 2017 competition. At this point, we might have achieved similar or better results than those obtained in 2017. Then, our team will try to develop its own architecture for this task by means of merging architectures or introducing new variants.

Notice that in order to obtain plain text, three steps must be followed to convert Fairseq predictions into text. As we have seen, if we wish to have plain text as an output in model's prediction, we will need to add some flags at the call of the model[4], otherwise, BPE decoding and detokenization must be implemented by us. Moreover, the preprocessing obtained with the WebNLG Baseline script requires that output predictions need to be relexicalise[5].

5 Evaluation

We are going to compare the performance of our models with respect to the benchmark. These performance metrics can be obtained in the generated paper in 2017 *The WebNLG Challenge: Generating Text from RDF Data* which contains the performance of proposed models for this competition. Therefore, we will use BLEU, TER and METEOR in order to evaluate and compare our models. Although Perplexity is not considered in The WebNLG Challenge report, we will study it because some recent papers suggest it for testing the generator.

The team could consider human evaluation, as an in-class activity, which can be discussed under the title of: Do machines generate coherent text? In that activity, we would like to ask for the quality of the generated text in terms of coherence. From this data, we could rank the models and see whether this rank differs from the obtained using mathematical metrics or not. Thus, providing notable insights on how are our models performing.

[2]The code implementation can be found in WebNLG Baseline GitLab.
[3]One code example of this process can be found here.
[4]Example of a predict call with decoding and detokenization: fairseq-interactive
[5]The code implementation can be found in WebNLG Baseline GitLab

6 Reference papers and benchmark

In the following, we present some literature written around the theme, as well as, the software documentation.

6.1 RDF-to-Text

This subsection includes from reports released in the WebNLG Challenge 2017 competition to new approaches considered after the competition.

- WebNLG Challenge by: Bayu Distiawan T
- Tilburg University models for the WebNLG challenge
- Triple-to-Text: Converting RDF Triples into High-Quality Natural Languages via Optimizing an Inverse KL Divergence
- Sequence-to-Sequence Models for Data-to-Text Natural Language Generation: Word- vs. Character-based Processing and Output Diversity
- GTR-LSTM: A Triple Encoder for Sentence Generation from RDF Data

6.2 Benchmark results

This paper summarises the results obtained in the WebNLG Challenge 2017 that will be considered as benchmark.

- The WebNLG Challenge: Generating Text from RDF Data

6.3 Software for developing

This is the documentation that we will address in order to develop the project.

- Facebook AI Research Sequence-to-Sequence Toolkit written in Python