Can a Transformer Assist in Scientific Writing? Generating Semantic Web Paper Snippets with GPT-2

Albert Meroño-Peñuela[1], Dayana Spagnuelo[1], and GPT-2[2]

[1] Vrije Universiteit, Amsterdam, The Netherlands
albert.merono@vu.nl, d.spagnuelo@vu.nl
[2] OpenAI Inc., San Francisco, California, US

Abstract. The Semantic Web community has produced a large body of literature that is becoming increasingly difficult to manage, browse, and use. Recent work on attention-based, sequence-to-sequence Transformer neural architecture has produced language models that generate surprisingly convincing synthetic conditional text samples. In this demonstration, we re-train the GPT-2 architecture using the complete corpus of proceedings of the International Semantic Web Conference since 2002 until 2019. We use user-provided sentences to conditionally sample paper snippets, therefore illustrating cases where this model can help at addressing challenges in scientific paper writing, such as navigating extensive literature, explaining the Semantic Web core concepts, providing definitions, and even inspiring new research ideas.

Keywords: Natural Language Generation · Semantic Web papers · Scholarly Communication

1 Introduction

A current scientific crisis revolves around the unmanageable pace at which new papers are being published. Studies show that over the past decades the number of published scientific papers has climbed by 8–9% each year; only in biomedicine 2 papers per minute are published in PubMed [6]. This causes problems to the traditional workflows of scientists, who lack resources for keeping up. The added load on an already resource-scarce scientific environment creates additional challenges: navigating scientific literature; writing papers; and getting new ideas becomes even harder. Moreover, humans have inherent limitations, such as not being systematic, introducing errors, having biases, and writing poor reports [3]. The use of AI to address these limitations has been identified as essential [5].

The Semantic Web, a research community that had its first international conference in 2002, is also exposed to these challenges. Only in 2019 its proceedings contained 1,377 pages and 569,371 words [4]; the complete 2002-2019 series contains 21,337,067 words. As time progresses, the entry cost to the knowledge and insights contained in these proceedings raises.

Language models have seen a spectacular improvement due to the introduction of deep neural architectures for long short-term memory [1]. Specifically, neural architectures based on the attention-based, sequence-to-sequence transformers such as BERT [2] and GPT-2 [8] have produced language models that generate surprisingly convincing synthetic conditional text samples. These models have been applied *e.g.*, to generate PubMed/MEDLINE abstracts[3] and investigate imaginary and unexplored hypotheses around climate change [7].

Here, we leverage the language learning and generation capabilities of GPT-2 for Semantic Web literature, and we re-train its small model (117M parameters) using the full corpus of International Semantic Web Conference (ISWC) proceedings. We focus on GPT-2 mainly due to its emphasis on auto-regression rather than the context of both sides of a word. Our goal is to investigate how AI and natural language generation can support the increasingly challenging task of writing Semantic Web papers. To do this, we first gather all ISWC proceedings volumes in PDF format, and we transform and prepare them in text form (Section 2). Then, we use this representation as training set for GPT-2, and we study the conditional samples it generates at given inputs (Section 3). We build a web-based interface on top of the model in order to demonstrate our approach (Section 4). Finally, we draw some conclusions and reflect on future work (Section 5).

2 Dataset

Our dataset is generated from the electronic version of the International Semantic Web Conference[4] (ISWC) proceedings. There are 18 proceedings ranging from the year of 2002, until 2019, with those after 2010 split into two parts due to their extensive length. This amounts into a total of 28 files processed by us.

We have converted each PDF file into TXT using the `pdftotext` command line tool. The tool can transcribe files while roughly maintaining their original physical layout, in the case of ISWC, the Lecture Notes in Computer Science (LNCS) template. Nonetheless, the tool is not precise, and introduces some conversion errors. These make the generated text, at times, meaningless to human readers. We have cleaned up most of these errors, and some other elements (*e.g.*, list of authors, table of contents, page headers) which disrupt the training of language models. In the following, we describe our data cleaning process.

2.1 Data cleaning

We clean the transcribed proceedings by leveraging from the LNCS template and its layout components. We use them to build regular expressions[5] which help us locate and remove unwanted content, in this particular order: 1. cover pages and meta information about the book; 2. running headers with authors

[3] https://twitter.com/DrJHoward/status/1188130869183156231

[4] Latest edition at time of writing: https://iswc2020.semanticweb.org/

[5] Script available at: https://github.com/dayspagnuelo/lncs_template_cleaner

and paper titles; 3. the list of organisation committee and sponsors, and the table of contents; 4. copyright footnotes; 5. list of references; and 6. author index. We also conduct some cleanings to help structuring better the output text, they clean some but not all instances of: 7. tables; 8. extra spaces and indentation (which also covers figures, algorithms, and formulae); and 9. extra lines. For the sake of brevity, we select a few components to give more explanation.

Cover pages. The initial pages of the proceedings contain the cover, and mostly information about the book and its printing. We remove everything found before the heading "Preface" as it does not contain scientific text on Semantic Web. We decided to include the preface in our dataset as it is also written by members of the Semantic Web community.

Running Headers. Running headers contain authors' names, paper titles and page numbers. They are the first content in most pages of the proceedings (except first page of each paper), which means they break the flow of natural text. We identify running headers by the presence of a *page break* character.

References. We remove references as they do not represent natural text. In order to remove the list of references of each paper we select content from the heading "References" until the next *page break* character.

The order of the cleaning steps is relevant. For instance, removing running headers before references. For short references sections that fit within one page, the next *page break* represents the beginning of the next paper. However, for longer ones taking more than one page, we must first remove the running headers, which also removes the page breaks characters in between the reference list.

Some automatic steps do not clean all the instances in a proceeding. One example are the Figures, which sometimes contain textual elements that get transcribed as disconnected words spaced in a random fashion. Building unambiguous regular expressions for those is challenging. We chose instead to search for extra spaces and indentation in the text, which normally includes figures and their captions. Some instances may remain, but the results of the last cleaning steps (7–9) help giving our dataset a natural text structure.

3 Training GPT-2 with Semantic Web Papers

Using the output of the clean ISWC text proceedings, we retrain GPT-2 using the 117M model as a basis in a 32 core Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz with 252GB of RAM. We use the framework provided directly by OpenAI[6]. First, we encode the dataset to prepare it for multiple runs (118s, 21,337,067 tokens). Then, we run the training process for 68,280 iterations and 332,533.83s (92.37h, average of 4.87s/it). The average loss at this point is 2.19 (with the last loss being 2.10). We then use the resulting model to generate conditional samples using various input texts. The following examples show the synthetic conditional samples generated by the model (in italics) at the provided input sentences (between quotes):

[6] https://github.com/openai/gpt-2

Example 1 (Explaining related work). "Reasoning with OWL ontologies at scale has proven to be a difficult problem in the Semantic Web. As shown by van Harmelen et al." *[1], the task of OWL reasoning involves the problem of identifying the relations between entities in a knowledge base. In this paper we present a series of approaches to the discovery of such relations. In particular, we focus on the problem of OWL reasoning by defining a set of axioms (or concepts) that can be used to classify knowledge bases.*

Example 2 (Definitions). "The Semantic Web is" *a rich and evolving web of interconnected resources with many different types of content, such as web pages, blogs, videos, music plays, and so on.*

Example 3 (Acknowledgements). See Acknowledgements below.

Example 4 (Explaining topics). "Decentralised querying on the Semantic Web with SPARQL has a number of advantages." *First, the presence of RDF graphs (i.e., named graphs) in the data, along with the possibility of querying them with SPARQL, allows the user to retrieve relevant data from the data cloud. Second, with the large amount of available data, the application is able to scale up to a large number of data sources.*

Example 5 (Suggesting novel ideas). "Many of the challenges of the Semantic Web have been partially solved using deep learning techniques based on embeddings. In this paper," *we present a system that generates a deep neural network (Deep Convolutional Network) from a set of neural networks and integrates it with a subset of the ontology that is used to model features of the text. Such neural networks are applied to model the textual content of the text, and the features are used to compute the semantic similarity between these features. The user-defined features can then be used to compute semantic features for the textual content of the text. Figure 2 shows the implementation of this class of features for the text in the system.*

Example 6 (Structured language). "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> SELECT * WHERE {" *?entity ?name . ?entity ?name . ?entity . ?entity . ?entity . ?entity . ?entity }*

Example 7 (Exploring research alternatives). "A number of different approaches investigate the problem of ontology matching in the Semantic Web. For example, " *the work on ontology matching by Euzenat [5] is an example of a model of matching. Recently, a new approach for ontology matching has been proposed that addresses the problem of ontology alignment based on the problem of ontology matching. The problem of ontology alignment based on similarity of concepts is described by similarity of classes and property.*

Example 8 (Paper structure). "The main contributions of this paper are" *as follows: – The first is a set of rules for the classification of relational data schemas. These rules are designed to be based on the following categories of data: – Datatypes: These are data that are the subject of at least one triple pattern and are considered to be an instance of schema. – Classes: These are either datatypes that are a subclass of data types, or datatypes that are not datatypes.*

Example 9 (Scientific hypotheses). "In this paper, our hypothesis is" *that a new ontology can be derived from an existing one. The key idea is to combine* *ontology-based and knowledge-based approaches, which are designed to provide reasoning and reasoning-based services that allow to perform reasoning tasks.*

Example 10 (Scientific research questions). "Therefore, our research question in this paper is" *how do we build a system that is able to learn and use such knowledge in a real context and that can answer this question in a way that is representative of a real problem.*

Example 11 (Providing explanations, Web-scale reasoning). "A simple explanation for the difficulty of solving large scale ontology reasoning is" *that we tend to solve small problems by imposing very big and complex rules. We often end up with very large portions of ontologies that cannot be represented using standard reasoners.*

Example 12 (Providing explanations, entity linking methods). "Machine learning techniques are used for the task of entity linking because" *it is a challenging task for the user. Therefore, we propose a novel method that is scalable to large knowledge bases with a high number of facts and a high accuracy.*

4 Demonstration

A web-based demonstrator of the trained model through conditional sampling is available at http://swgpt2.amp.lod.labs.vu.nl/. As parameters for conditional sampling, we set the temperature at 1 and the diversity at 40. After the service loads the required libraries, a `Model prompt` is displayed. The user can then type the sentence, followed by the enter key, that will be used as input to the model for conditional sampling. After a few seconds, the model outputs a sample.

The demonstration on the floor will make use of this prompt for conditional sampling. Users will be instructed to provide contexts of various lengths, as well as finished and unfinished sentences. The guidance for the input sentences, as well as the generated content, will include: (a) Semantic Web topics (*e.g.*, knowledge graph construction, querying, ontologies, APIs, reasoning, etc.); (b) Structured and unstructured content (*e.g.*, RDF vs natural language); (c) Outlines, citations, and other scholarly features; (d) Well-known authors in the community.

5 Conclusion and Future Work

In this paper, we describe a demonstration that uses the GPT-2 transformer architecture to learn a language model for a cleaned corpus of 2002–2019 ISWC proceedings, and leverages this model to generate samples conditioned on input. The demonstration is available as a public Web interface. Our findings are that the model can be used to generate meaningful texts that can be used for various scientific writing tasks, such as explaining related work, providing definitions, or proposing hypotheses. We think this work can be used for scientific writing assistance, as well as inspire new research directions through human-machine brainstorming.

From the social perspective, we are aware of the ethical implications of using natural language generation and AI for scientific writing, including the need for accountability, the shared responsibility of all contributors (humans or machines), and the requirement on these contributors to fully understand what they report on. In this sense, we see this work more as an assistance and a tool for human writers, as seen in *e.g.*, Gmail's auto-complete, rather than a substitute.

We foresee various possibilities to continue this work in the future. First, we plan to to retrain GPT-2 adding the whole collection of ESWC papers, increasing the scale of the experiment and testing the robustness of our dataset cleaning strategy. Second, we will investigate methods for dynamically generating this cleaning strategy, and reusing the training set for different user-specific goals. Third, we want to expand our approach by leveraging the knowledge already available in Knowledge Graphs, and generate the seed conditioning sentences by querying Knowledge Graphs to effectively guide the text generation through real-world models and semantic pathways. Fourth, we plan on using this language model for downstream tasks other than text generation, *e.g.*, finding similar papers by using alternative wordings.

Acknowledgements. "This paper would not have been possible without the support of several persons and institutions. GPT-2 would like to thank" *all of the members of the GPT-1 technical committee.* The authors want to thank Frank van Harmelen, Paul Groth, and the anonymous reviewers for their valuable comments. This work is partly supported by the CLARIAH project funded by NWO.

References

1. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078 (2014)
2. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), http://arxiv.org/abs/1810.04805
3. Garijo, D., Kinnings, S., Xie, L., Xie, L., Zhang, Y., Bourne, P.E., Gil, Y.: Quantifying reproducibility in computational biology: the case of the tuberculosis drugome. PloS one **8**(11) (2013)
4. Ghidini, C., Hartig, O., Maleshkova, M., Svatek, V., Cruz, I., Aidan, H., Song, J., Lefrançois, M., Gandon, F.: The Semantic Web-ISWC 2019. Springer (2019)
5. Gil, Y.: Thoughtful artificial intelligence: Forging a new partnership for data science and scientific discovery. Data Science **1**(1-2), 119–129 (2017)
6. Landhuis, E.: Scientific literature: information overload. Nature **535**(7612), 457–458 (2016)
7. Pearce, W., Niederer, S., Özkula, S.M., Sánchez Querubín, N.: The social media life of climate change: Platforms, publics, and future imaginaries. Wiley Interdisciplinary Reviews: Climate Change **10**(2), e569 (2019)
8. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners. OpenAI Blog **1**(8), 9 (2019)