NILC at SR'20: Exploring Pre-Trained Models in Surface Realisation

Marco Antonio Sobrevilla Cabezudo msobrevillac@usp.br
Thiago Alexandre Salgueiro Pardo taspardo@icmc.usp.br

Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo
Avenida Trabalhador São-carlense, 400. São Carlos - SP - Brazil

Abstract

This paper describes the submission by the NILC Computational Linguistics research group of the University of São Paulo/Brazil to the English Track 2 (closed sub-track) at the Surface Realisation Shared Task 2020. The success of the current pre-trained models like BERT or GPT-2 in several tasks is well-known, however, this is not the case for data-to-text generation tasks and just recently some initiatives focused on it. This way, we explore how a pre-trained model (GPT-2) performs on the UD-to-text generation task. In general, the achieved results were poor, but there are some interesting ideas to explore. Among the learned lessons we may note that it is necessary to study strategies to represent UD inputs and to introduce structural knowledge into these pre-trained models.

1 Introduction

Universal Dependencies[1] (UD) have gained relevance in the Natural Language Processing (NLP) community. UD treebanks have already proved useful in the development of multilingual applications, becoming an advantage for developers.

One of the successful applications of UD is related to Data-to-text generation. This may be seen in the two shared-tasks proposed (Mille et al., 2018; Mille et al., 2019) in which there were several participants. In general, the Surface Realisation Shared Task aims to continue with the development of natural language generation methods focused on the surface realisation task. Specifically, models in this task must generate sentences from dependency trees (or similar structures) in CoNLL format.

This task comprises two tracks: (1) the Shallow Track, in which word order and word forms are removed from the UD structure, and (2) the Deep Track, which, in addition to the word order and word forms, functional words, morphological features and other kinds of information are removed or changed in the UD structure to make it more similar to a semantic representation (called deep syntax).

Diverse approaches have been applied on this shared task in previous editions, being mainly divided in inflection generation and word ordering tasks (Mille et al., 2019). Besides, there is a trend to use neural models for the same tasks.

Recently, pre-trained language models have become standard in a variety of Natural Language Processing (NLP) tasks (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019), including sentence-level classification, sequence tagging and question answering. These models can be pre-trained on large corpora of available unannotated text, and then fine-tuned for specific tasks on smaller amounts of supervised data, relying on the induced language model structure to facilitate generalisation beyond the annotations.

These pre-trained models have been widely used in text-to-text generation tasks such as text simplification, automatic summarisation, and machine translation, obtaining good results and outperforming the current state of the art. However, there are few initiatives for the data-to-text generation task. Lately, Mager et al. (2020) used GPT-2 for fine-tuning AMR-to-text generation task, showing improvements and

[1] Available at https://universaldependencies.org/

Proceedings of the Third Workshop on Multilingual Surface Realisation (MSR'20), pages 50–56
Barcelona, Spain (Online), December 12, 2020.

that current pre-trained models can handle these representations even if the knowledge is not explicitly structured.

In this context, this paper presents the description of the system submitted by the NILC team to the track 2 of the Surface Realisation Shared Task 2020 (Track 2 - SR'20) (Mille et al., 2020). Our proposal is an End-to-End approach inspired by the work of Mager et al. (2020). We explore some strategies to sequentially represent UD structures and to fine-tune GPT-2 (Radford et al., 2019) on the pre-processed dataset[2].

2 Track 2 Dataset - SR'20

The dataset for the Track 2 is composed by UD structures and their corresponding sentences. The UD structure is similar to a dependency tree, however, some information are modified:

- word order is removed by randomised scrambling;
- words are replaced by their lemmas;
- some prepositions and conjunctions (that can be inferred from other lexical units or from the syntactic structure) are removed;
- determiners and auxiliaries are replaced (when needed) by attribute/value pairs;
- edge labels were generalised into predicate argument labels in the PropBank/NomBank fashion;
- morphological information coming from the syntactic structure or from agreement is removed;
- only coarse-grained Part-of-Speech tags are kept.

Figure 1 and 2 show the CoNLL and graphic representation for the sentence "Two of them were being run by 2 officials of the Ministry of the Interior!". In Figure 1, we may see "*idX*" and "*original_id*" attributes, where "X" can be a number. This attributes are related to the track 1 and the original ids (positions) of the tokens in the sentence and are removed from the test set.

```
run _    VERB     _    Tense=Pres|id2=8|id1=12|id3=13|original_id=6|Aspect=Prog|ClauseType=Exc 0   ROOT _  _
official     _    NOUN     _    Number=Plur|id2=3|id1=16|original_id=9  1   AM  _   _
two _    NUM _    NumType=Card|id1=6|original_id=1    1   A2  _   _
Ministry     _    PROPN    _    Number=Sing|id2=14|id1=11|id3=15|original_id=12|Definite=Def    2   AM  _   _
2    _    NUM _    NumType=Card|id1=10|original_id=8   2   A1INV   _   _
they     _    PRON     _    Number=Plur|id2=2|id1=4|original_id=3|Person=3|PronType=Prs 3   AM  _   _
Interior     _    PROPN    _    Number=Sing|id2=7|id1=5|id3=9|original_id=15|Definite=Def   4   AM  _   _
```

Figure 1: Deep track example ("Two of them were being run by 2 officials of the Ministry of the Interior!") in CoNLL format.

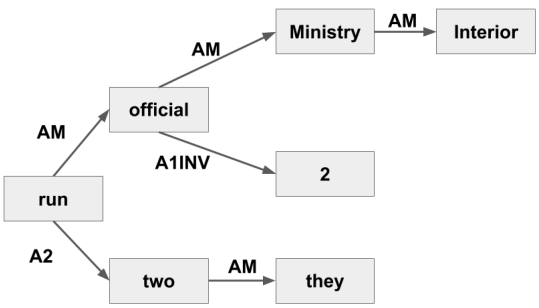

Figure 2: Representation of the example in graphic format.

[2]The corresponding source code is available at https://github.com/msobrevillac/pretrained-amr-to-text

Finally, the dataset contains subsets from different domains. For English (our target language in this task), there are 4 files in the training and development (dev) set each, and 7 files for the test set from the previous edition (Mille et al., 2019) and 1 file for the test set in this edition.

3 System Description

In first edition of this shared task, we submit a system in which we use a data augmentation strategy (Sobrevilla Cabezudo and Pardo, 2018) to deal with the track 1. However, for this edition, we focus on the track 2 and use only resources allowed by the shared task (closed sub-track). Differently from most of the work found in the literature, we propose an End-to-End approach, jointly learning the inflection generation and word ordering tasks.

3.1 GPT-2 for UD structures

Inspired by the work of Mager et al. (2020), we use GPT-2 and fine-tune on the joint distribution of UD structure and text. Given a tokenized sentence $w_1 w_N$ and the sequential UD structure $a_1 ... a_M$, we maximize the joint probability.

$$p_{GPT-2}(w, a) = \prod_{j=1}^{N} p_{GPT-2}(w_j | w_{1:j-1}, u_{1:M}) \prod_{i=1}^{M} p_{GPT-2}(u_i | u_{1:i-1}) \quad (1)$$

A special separator token is added to mark the end of the sequential UD structure. Relations that should not be interpreted literally are assigned tokens from the GPT-2 unused token list (adding a ":" to mark the token as a relation). Furthermore, in the case of morphological information, values in feature name-value pairs are considered common tokens and feature names are considered relations. For example, in Figure 1, the token "run" has "Tense=Pres" as a feature name-value pair. This way, "Pres" is considered a common token and "Tense" a relation. At test time, we provide the UD structure as context as in conventional conditional text generation.

It is worth noting that we explore different ways to build the sequential (linearised) UD structure, and all these sequential versions are derived from the PENMAN notation. We explore the following linearised versions:

(A) PENMAN format: this format is the same one used in Abstract Meaning Representation (AMR);

(B) PENMAN format without morphological relations: the same format, but removing all morphological relations (and others in the same column), such as "Tense" and "Aspect", among others;

(C) PENMAN format without morphological relations and parentheses;

(D) PENMAN format without parentheses: the same as the first one but removing the parentheses;

(E) PENMAN format without relations: the same as the first one but removing all the relations;

(F) PENMAN format without relations and parentheses: the same as the first one but removing all the relations and parentheses.

Figure 3 shows three representations for the example in Figure 1: (A) PENMAN notation, (B) PENMAN notation without morphological relations, and (C) PENMAN notation without morphological relations and parentheses. It is interesting to add that the parentheses in PENMAN notation provide information about the graph structure of the input.

```
(A)  ( run :tense pres :aspect prog :clausetype exc
         :am ( official :number plur
               :am ( Ministry :number sing :definite def
                     :am ( Interior :number sing :definite def ) )
               :a1inv ( 2 :numtype card ) )
         :a2 ( two :numtype card
               :am ( they :number plur :person 3 :prontype prs ) ) )

(B)  ( run pres prog exc
         :am ( official plur
               :am ( Ministry sing def
                     :am ( Interior sing def ) )
                     :a1inv ( 2 card ) )
         :a2 ( two card
               :am ( they plur 3 prs ) ) )

(C)  run pres prog exc :am official plur :am Ministry sing def :am Interior sing def :a1inv 2 card
     :a2 two card :am they plur 3 prs
```

Figure 3: Sequential UD structures for the sentence "Two of them were being run by 2 officials of the Ministry of the Interior!". (A) PENMAN notation, (B) PENMAN notation without morphological relations, and (C) PENMAN notation without morphological relations and parentheses.

3.2 Settings

We use the small GPT-2 model provided by HuggingFace (Wolf et al., 2019)[3]. The model is trained on the joint of all training subsets. The fine-tuning is executed in 7 epochs (as the model converges at this time), using a batch size of 8, the AdamW optimizer with a learning rate of 6.25e-5, a max length of 350 in the source and target, and freezing the embeddings. For the decoding, we use a beam size of 15.

At test time, we get tokenised sentences. We then post-process them by using the Moses detokeniser[4].

4 Results and Discussion

The automatic performance of the diverse proposals at the shared task is computed by the following measures:

- BLEU (Papineni et al., 2002): precision metric that computes the geometric mean of the n-gram precisions between the generated and reference texts, adding a brevity penalty for shorter sentences (we use the smoothed version and report results for n = 1, 2, 3, and 4);
- NIST (Doddington, 2002): related n-gram similarity metric weighted in favor of less frequent n-grams, which are taken to be more informative;
- Normalized edit distance (DIST): inverse, normalized, character-based string-edit distance that starts by computing the minimum number of character insertions, deletions and substitutions (all at cost 1) required to turn the system output into the (single) reference text;
- BertScore (Zhang et al., 2020): leverages the pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity.

Table 1 shows the results of the different linearised versions (described in Section 3.1 of the UD structures on the development set). The results are the ones obtained on the join of all dev subsets provided by the shared-task.

In general, morphological relations do not seem to be necessary as the performance improves when these are removed (B in Table 1). However, a possible noise in this analysis could be generated by

[3]We use only the small GPT-2 version as we could not execute this on our current server.

[4]We use the perl code available at `https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer`

the input length since including morphological relations (linearised version A in Table 1) could make the input length larger and the max length parameter could delete some important tokens, resulting in a lower performance in comparison with linearised version B.

To make the analysis of omitting morphological relations clearer, we may see versions C (without parentheses and morphological relations) and D (without parentheses). Both versions contain fewer tokens (in relation to A and B versions) and one may see that disregarding morphological relations produces improvements (results in version C are better than in D).

Another point to note is that parentheses are the most important tokens as they represent the structure of the input. Therefore, removing them from the input leads to a significant drop in the performance (D). Furthermore, this drop is bigger than the one obtained by leaving out all the relations (E), showing that parentheses could encode some information about the relations among the nodes. Finally, the results of the F version could suggest that, although parentheses encode some information about the relations, there are more information that is not encoded, making the use of relations (in general) necessary.

Linearised version	BLEU	NIST	BertScore
(A)	39.28	9.65	0.9390
(B)	42.52	10.29	0.9413
(C)	36.67	9.82	0.9350
(D)	35.89	9.59	0.9347
(E)	38.00	9.97	0.9354
(F)	27.61	8.83	0.9228

Table 1: Results with different inputs on the dev set.

Table 2 shows the results of the automatic evaluation on several test sets. The model that we use to get these results is the one that got the best results in the dev set. In general, Table 2 shows that all values are close to the average (except for the test set presented in this edition). This could suggest that GPT-2 can keep a similar performance across the domains, i.e., it can generalise well. Finally, our approach got a performance lower than other approaches for the same track. However, we expect that these differences could be reduced if we use a bigger model such as medium or large GPT-2 (similar to the results of Mager et al. (2020)).

Table 3 shows the results of the human evaluation on two test sets predefined by the organizing committee. Specifically, Direct Assessment (Graham et al., 2017) is applied to conduct this evaluation. Candidate outputs are presented to human assessors, who rate their (i) meaning similarity (relative to a human-authored reference sentence) and (ii) readability (no reference sentence) on a 0-100 rating scale. The metric used for ranking the different systems is the average standardised score (avg. z in Table 3. We may see that our approach still have problems representing the correct reference as this gets the worst performance according to the meaning similarity (last cluster). However, when readability is evaluated,

		BLEU	NIST	DIST	BertScore
2020	en_filtered_lines-fix	39.68	9.31	57.56	0.9317
	en_pud	42.60	9.68	59.85	0.9382
	en_ewt-Pred-HIT-edit	43.15	9.64	62.20	0.9392
	en_pud-Pred-LATTICE	42.64	9.59	60.43	0.9368
2019	en_ewt	45.19	9.96	64.83	0.9411
	en_gum	40.94	9.00	60.42	0.9399
	en_lines	41.04	9.09	61.18	0.9381
	en_partut	43.41	8.24	59.74	0.9438
Average		42.33	9.31	60.78	0.9386

Table 2: Results of our system on the test set.

Meaning Similarity							Readability						
Test set	Team	Sub-track	Avg.	Avg. z	n	N	Test set	Team	Sub-track	Avg.	Avg. z	n	N
English (EWT)	IMS	20b	85.1	0.272	1,667	1,927	English (EWT)	Concordia	20a	71.8	0.321	806	908
	IMS	20a	84.7	0.259	1,701	1,942		Ours	20a	68.6	0.185	823	947
	Concordia	20a	84.7	0.245	1,675	1,897		IMS	20b	67.3	0.159	807	936
	IMS	19	82.7	0.201	1,692	1,920		IMS	20a	65.8	0.109	753	866
	Ours	20a	75.6	-0.079	1,657	1,892		IMS	19	63.6	0.027	808	923
English (Wiki)	IMS	20b	87.3	0.157	700	1,016	English (Wiki)	Concordia	20a	80.6	0.37	952	1,283
	IMS	20a	85.6	0.057	755	1,078		Ours	20a	75.4	0.213	930	1,273
	IMS	19	85.5	0.025	698	1,023		IMS	20b	70.2	0.055	932	1,256
	Concordia	20a	84.7	-0.029	715	1,036		IMS	20a	69	-0.03	963	1,284
	RALI	19	76	-0.463	720	1,044		IMS	19	67.3	-0.095	932	1,233
	Ours	20a	76.6	-0.491	721	1,088		RALI	19	56.1	-0.562	940	1,329

Table 3: Results of the human evaluation for the track 2. Meaning Similarity and Readability are computed. Avg. is the average 0-100% received by systems. Avg. z is the corresponding average standardised scores. "n" is the total number of distinct test sentences assessed, and N is the total number of human judgments. The results are sorted by avg. z. and horizontal lines indicate clusters, such that systems in a cluster all significantly outperform all the systems in lower ranked clusters.

we obtain the second best results (second cluster), even compared with approaches in the open subtrack (20b) in which all kinds of resources are allowed.

These results are expected as the automatic evaluation shows a low performance for our approach and it is reflected in the meaning similarity evaluation, while GPT-2 is a robust language model and knows how to build coherent sentences (we have to stress that readability is evaluated without references). More experiments could be done in order to explore how to get improvements in meaning similarity. Experiments performed by Mager et al. (2020) show that the performance improves significantly when a bigger version of GPT-2 is used. Besides, we may see that the performance varies widely according to the linearisation strategy, which would be an interesting research line to explore in the future.

5 Conclusion and Future Work

This paper describes the application of a pre-trained model, the GPT-2, to the UD-to-text generation task in the context of the Surface Realisation shared task. Results show that the way as the UD structures are linearised is important for the model in this task. Thus, an interesting research line for future work could be investigating other ways to represent/linearise UD structures and to introduce the knowledge about structure in this kind of model. As future work, we also plan to apply this approach to other languages and use a bigger version of GPT-2.

Acknowledgements

This work was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES)* – Finance Code 88882.328822/2019-01. The authors are also grateful to USP Research Office (PRP 668) for supporting this work, and would like to thank NVIDIA for donating the GPU. This work is part of the OPINANDO project (`https://sites.google.com/icmc.usp.br/opinando/`) and the USP/FAPESP/IBM Center for Artificial Intelligence (C4AI - `http://c4ai.inova.usp.br/`). Finally, this research is carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP (grant 2013/07375-0).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Manuel Mager, Ramón Fernandez Astudillo, Tahira Naseem, Md Arafat Sultan, Young-Suk Lee, Radu Florian, and Salim Roukos. 2020. GPT-too: A language-model-first approach for AMR-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1846–1852, Online, July. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (SR'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12, Melbourne, Australia, July. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, and Leo Wanner. 2019. The second multilingual surface realisation shared task (SR'19): Overview and evaluation results. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 1–17, Hong Kong, China, November. Association for Computational Linguistics.

Simon Mille, Anya Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham, and Leo Wanner. 2020. The third multilingual surface realisation shared task (SR'20): Overview and evaluation results. In *Proceedings of the 3nd Workshop on Multilingual Surface Realisation (MSR 2020)*, Dublin, Ireland, December. 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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

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

Marco Antonio Sobrevilla Cabezudo and Thiago Pardo. 2018. NILC-SWORNEMO at the surface realization shared task: Exploring syntax-based word ordering using neural models. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 58–64, Melbourne, Australia, July. Association for Computational Linguistics.

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. *ArXiv*, abs/1910.03771.

Tianyi Zhang, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.