Surface Realization Using Pretrained Language Models

Farhood Farahnak and **Laya Rafiee** and **Leila Kosseim** and **Thomas Fevens**
Concordia University
Montreal, Canada
`firstname.lastname@concordia.ca`

Abstract

In the context of Natural Language Generation, surface realization is the task of generating the linear form of a text following a given grammar. Surface realization models usually consist of a cascade of complex sub-modules, either rule-based or neural network-based, each responsible for a specific sub-task. In this work, we show that a single encoder-decoder language model can be used in an end-to-end fashion for all sub-tasks of surface realization. The model is designed based on the BART language model that receives a linear representation of unordered and non-inflected tokens in a sentence along with their corresponding Universal Dependency information and produces the linear sequence of inflected tokens along with the missing words. The model was evaluated on the shallow and deep tracks of the 2020 Surface Realization Shared Task (SR'20) using both human and automatic evaluation. The results indicate that despite its simplicity, our model achieves competitive results among all participants in the shared task.

1 Introduction

Natural Language Generation (NLG) models aim to generate fluent human-like texts given structured data. This involves both *content planning* (selecting the content to communicate) and *surface realization* (selecting, ordering, and inflecting the actual words) (Hovy et al., 1997; Reiter and Dale, 2000). This paper focuses on the second sub-task: Surface Realization (SR).

Unlike many tasks in Natural Language Processing (NLP), the performance of SR models are still below human performance. In order to fill this gap and encourage more research in this field, several shared tasks in NLG and SR have been proposed. In particular, since 2018, the Surface Realization Shared Tasks (Mille et al., 2018; Mille et al., 2019; Mille et al., 2020) were introduced to provide common-ground datasets for developing and evaluating NLG systems. This year, the task (Mille et al., 2020) proposed two tracks in several languages including English: 1) Track1: *shallow track* and 2) Track2: *deep track*. In the shallow track, unordered and lemmatized tokens with Universal Dependency (UD) structures (de Marneffe et al., 2014) were provided to participants and systems were required to reorder and inflect the tokens to produce final sentences. The deep track was similar to the shallow track but functional words and surface-oriented morphological information were not provided and had to be inferred by the systems. Therefor, in addition to determining the order and the inflection of tokens, systems participating in the deep track had to guess the omitted words.

Considering that the input data is in the form of Universal Dependency (UD) structure, data-to-text models, and graph-to-text models in particular, seem to be the right choice for the task of surface realization. However, in this study, we take a different path to tackle the problem. Our proposed model is designed based on text-to-text approaches using a pretrained encoder-decoder language model. More specifically, a BART (Lewis et al., 2020) language model is used for the task of surface realization for both the shallow and the deep tracks. The proposed approach is an end-to-end model trained on a linearized representation of the graph of the sentences with their corresponding Universal Dependency information. The results on the English datasets demonstrate the potential of these models for surface realization tasks and in general data-to-text problems.

In the following sections, we first review recent and related approaches in surface realization and text-to-text models, then in Section 3, our proposed model is explained. Section 4 describes the results of our model on the surface realization datasets. Finally, Section 5 discusses conclusion and future work.

2 Related Work

Surface realization typically involves three tasks: syntactic realization, morphological realization, and orthographic realization (Reiter and Dale, 2000). Syntactic realization tries to identify the proper ordering of the input data, whereas morphological and orthographic realization are responsible for word inflections, punctuation, and formatting.

Several surface realization models presented at the previous Surface Realization Shared Task (Mille et al., 2019) used a cascade of pointer-based models for syntactic realization followed by another neural network module for morphological and orthographic realization (Du and Black, 2019; Farahnak et al., 2019; Mazzei and Basile, 2019). For example, Du *et al.* (2019) utilized a graph attention network (GAT) (Veličković et al., 2018) for encoding the input sentences and a pointer decoder (Vinyals et al., 2015) to select the next element from their graph. Whereas Yu *et al.* (2019) used a bidirectional Tree-LSTM (Zhou et al., 2016) as the encoder and an LSTM (Hochreiter and Schmidhuber, 1997) as the decoder and multiple LSTM modules for morphological and orthographic realization tasks. These two approaches achieved the highest performance among all participating systems at SR'19.

On the other hand, when the text to generate is conditioned on the content provided in the form of graphs, tables, etc then data-to-text generation models are utilized. As a family of data-to-text models, graph-to-text generation tries to generate natural text given its input graph. Graph-to-text generation models employ graph encoders to obtain a suitable representation from the input graph. Several applications such as text summarization (Duan et al., 2017), question answering (Fan et al., 2019), as well as surface realization (Du and Black, 2019) have used these types of generation models.

Recently, several works have proposed to use language models in the form of text-to-text for what is inherently data-to-text and particularly graph-to-text tasks (Kale, 2020; Mager et al., 2020; Harkous et al., 2020). Instead of modeling the node and edges of the graphs, they mapped the graph structure as sequences of words and let the language model encode the graph information. Kale (2020) takes advantage of T5 (Raffel et al., 2019) for data-to-text problems, whereas Mager *et al.* (2020) and Harkous *et al.* (2020) utilize GPT2 (Radford et al., 2019).

Following this recent trend of text-to-text models for graph-based problems, we developed an end-to-end approach based on a language model for the problem of data-to-text generation. The approach maps the given Universal Dependency structures to surface forms to tackle all the tasks required for the surface realization.

3 Model

Following the success of pretrained language models for data-to-text generation (Kale, 2020; Harkous et al., 2020), we tackled the surface realization problem using a similar approach. BART (Lewis et al., 2020) is an encoder-decoder language model based on transformers (Vaswani et al., 2017). Specifically, BART is a denoising autoencoder model trained on several denoising tasks making it applicable for a variety of downstream NLP tasks. The language model is trained to first encode the input, and then generate the text based on its input representation. The encoder-decoder architecture of BART makes it suitable for our task where the encoder module encodes the graph representation into an embedding space then the decoder generates the inflected form of the input one token at each decoding step. Hence, the model performs syntactic, morphological, and orthographic realization all at the same time.

In order to take advantage of a pretrained language model, BART in our case, we first need to map the graph structure of the input into a linear representation (i.e. plain text). For this task, given all the Universal Dependency (UD) information provided by the SR'20 organizers, we considered using only `LEMMA`, `UPOS`, `FEATS`, `HEAD`, and `DEPREL` for each node. Figure 1 shows the mapping structure with

```
       ...
(a)    26     kill    killed  VERB    VBD     Mood=Ind|Tense=Past|VerbForm=Fin|original_id=7  20      parataxis       _       _
       ...

(b)                           <s> LEMMA < HEAD > UPOS | FEATS | DEPREL <\s>

(c)                       ... <s> kill < Al > VERB | Ind Past Fin | parataxis <\s> ...
```

Figure 1: A sample token with its linearized representation. The sample token with index 26 and its Universal Dependency (UD) information in (a) will be encoded using (b) to generate its linearized representation in (c).

an example of a linearized node. As Figure 1 shows, we concatenate each token with its corresponding features where special tokens `<`, `>`, `|` identify the boundary of each feature and special tokens such as the beginning of sentence `<s>` and end of sentence `<\s>` determine the boundary of each node (i.e. token). In these mapped representations, instead of the index for the `HEAD` feature, the actual token of the `HEAD` is used.

As opposed to previous work where tokens are selected among the input (Du and Black, 2019; Mazzei and Basile, 2019), the BART decoder generates tokens at each decoding step. As a result, we have no control on the number of tokens that should be generated. This can lead the model to generate extra tokens or ignore some of the input tokens. To alleviate this issue, we considered the tokenized form of the target sentences as the targets of the model. This allows the model to learn to map each node representation in its input into a token in the output and achieve a 1:1 mapping between the number of nodes in the input and the number of output tokens. An alternative would be to train on the detokenized output form but for the sake of automatic evaluation of the shared task, we did not follow this option.

4 Experiments and Results

4.1 Datasets

The proposed text-to-text surface realizer model was evaluated on the English datasets of the SR'20 (Mille et al., 2020). The datasets were created using the Universal Dependency (UD) datasets (de Marneffe et al., 2014) where the tokens within each sentence were randomly shuffled and the inflections were removed. The training and development sets were accompanied by 7 features. Out of these, the `FEATS` feature contained the relative linear order with respect to the governor (`Lin`) in addition to more than 40 morphological sub-features from the universal feature inventory. Among the four English datasets provided for training, we only trained on `en_ewt-ud-train` while the performance of the model was measured on all the eight test sets provided by the organizers (see Table 1).

4.2 Model Configuration

We used the pretrained BART-large model provided in the Huggingface library (Wolf et al., 2019). This model has 12 layers in each encoder and decoder modules. For each task, we trained our models with the same cross-entropy loss function as suggested in the original paper of BART and AdamW algorithm (Loshchilov and Hutter, 2019) with batch size of 2 and the learning rate of $1e-5$ for 20 epochs where these hyper-parameters are chosen based on the development set. For efficiency in training time, sentences longer than 35 tokens were removed from the training set. During inference, we used beam search with a size of 5.

4.3 Results

Our surface realization model was evaluated on all seven test sets from SR'19 and one new test set (`en_Wikipedia`) for SR'20. Results with all test sets are presented in Table 1 where *in-domain* refers to set-ups where test sets and training sets are from the same domain, while *out-of-domain* refers to set-ups where test sets are not in the same domain as the training sets. Two types of test sets were provided: *ud* and *pred*. The *ud* test sets contain the original UD datasets; whereas the *pred* test sets contain automatic predicted parse trees. Since we only trained our model on `en_ewt-ud-train`, all the other test sets should be considered out-of-domain for our model.

			T1				T2			
#		Dataset	BLEU	NIST	DIST	BERTScore	BLEU	NIST	DIST	BERTScore
1	In-domain	`en_ewt-ud`	70.71	12.70	77.94	0.9565	58.44	11.61	73.66	0.9635
2		`en_ewt-Pred`	67.12	12.40	74.47	0.9562	56.01	11.19	69.00	0.9593
3	Out-of-domain	`en_pud-ud`	74.47	12.62	76.46	0.9605	**58.45**	11.43	68.83	**0.9613**
4		`en_pud-Pred`	73.41	12.52	77.68	0.9596	**56.69**	11.18	67.79	**0.9584**
5		`en_gum-ud`	66.98	11.62	69.87	0.9555	53.92	10.51	67.02	0.9572
6		`en_lines-ud`	62.70	11.30	68.62	0.9565	47.96	9.93	64.33	0.9487
7		`en_partut-ud`	67.05	9.83	71.59	0.9565	**50.54**	8.57	62.39	0.9505
8		`en_Wikipedia`	74.66	12.90	76.93	0.9614	**57.49**	11.57	67.26	0.9473

Table 1: Results of our submission in the shallow (T1) and deep (T2) tracks of SR'20. Highlighted values indicate the highest score among all participants achieved by our model.

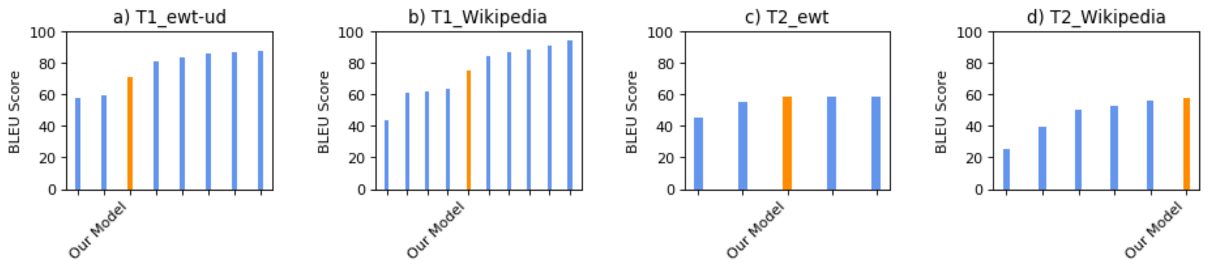


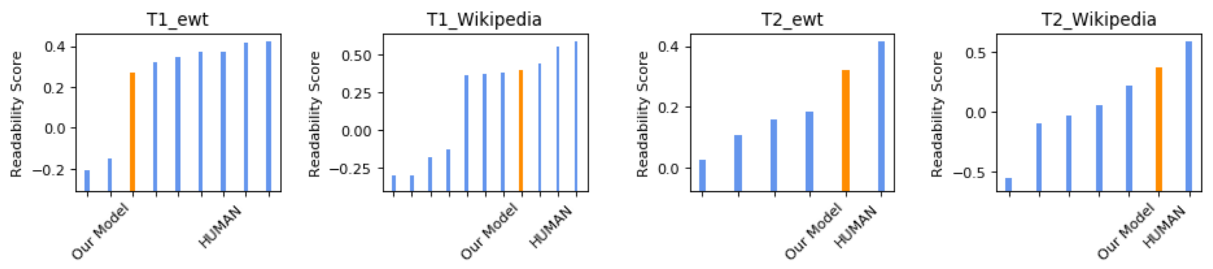
Figure 2: Comparison of all systems submitted to SR'20 on the `en_ewt-ud` and `en_Wikipedia` test sets for the shallow (T1) and the deep (T2) tracks evaluated based on their BLEU score.

Our models were evaluated both with automatic metrics and human evaluation. For the automatic metrics, as shown in Table 1, each test set was evaluated using four different metrics: BLEU, NIST, DIST, and BERTScore.

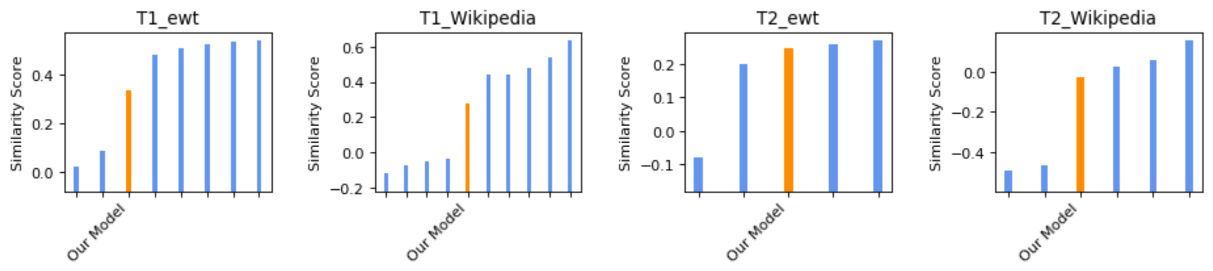
Figure 2 compares the BLEU score of our model with all other participants with the test sets `en_ewt-ud` and `en_Wikipedia` for both shallow (T1) and deep (T2) tracks. As Figure 2 shows, our model's performance falls close to the median of the other models in the shared task for the shallow track. However, in the deep track, our model performs best (Figure 2d) or close to the best (Figure 2c). These results seem to indicate that BART can better predict functional words and surface-oriented morphological information (required only in deep track) and this is mainly due to the denoising approaches used for training BART.

For human evaluation, SR'20 used Direct Assessment (Graham et al., 2017) where human assessors rated the output of systems based on meaning similarity (relative to a human-authored reference sentence) and readability (without reference sentences). Figure 3 shows the average score for the meaning similarity and readability on both shallow and deep tracks on two test sets: `en_ewt-ud` nad `en_Wikipedia`. In terms of readability, in the deep track, our model achieved the highest score after Human (see Figure 3a). However, in the shallow track, our performance is at the median of all participants. On the other hand, the meaning similarity of our model is lower than the best performing model (see Figure 3b). These comparisons indicate that although our model is able to generate well-written texts, their meaning are not close enough to the target sentences.

In order to have a better understanding of the model's behaviour, we further analyzed the generated outputs of our model. Our decoder generates tokens from its encoder hidden space, which is in contrast with ordering models where the output is selected from their input. This characteristic can have two impacts: A) the model does not restrict the output to the words in the input and it has no control over the number tokens it should generate. As a result, the model can generate extra tokens or ignore tokens from the input. Example A in Table 2 highlights this issue where the model did not generate the token *Ali*. B) the model generates long tokens in more than a single decoding step which makes it too complex for the model to have control over the details of such tokens. This behaviour is shown in Table 2 Example B. In this example, even though the model was able to generate a very complex URL similar to its input, there



(a) Readability



(b) Meaning Similarity

Figure 3: Human evaluation comparison of systems submitted in SR'20 on the `en_ewt-ud` and `en_Wikipedia` test sets for shallow (T1) and deep (T2) tracks evaluated based on their (a) readability and (b) meaning similarity.

A	Target	Shazad Khuram Ali , 27 , High Wycombe
	T1 output	Shazad Khuram , 27 High Wycombe ,
	T2 output	Shazad Khuram 27 High Wycombe Ali
B	Target	I hope you do n't mind , but I 've taken liberty to turn them into a web photo album at http://24.27.98.30/pictures/08-05_Garrett_Gayle_Bday .
	T1 output	I hope you do n't mind , but I 've taken liberty to turn them into a web photo album at http://24.98.30/pictures/08-05_garrett_gayle_bday .
	T2 output	I hope you do n't mind but I have taken liberty to turn them into a web photo album (http://24.98.30/pictures/08-05_garrett_gayle_bday.html) .

Table 2: Examples of unsuccessful outputs generated by our model on the shallow (T1) and deep (T2) tracks.

are some missing parts such as *27* or even extra parts such as *.html*. It should be noted that in Example B, the entire URL is considered as a single token.

5 Conclusion

In this work, we used a language model (text-to-text) to tackle the graph-to-text surface realization problem. We showed that using a pretrained encoder-decoder language model such as BART, can allow the model to reconstruct the tokens' orders and inflections in an end-to-end fashion without any modification in the network architecture. This shows that the pretrained language model is able to encode the graph information and map this representation into a human readable text. The results of the deep track showed that our proposed model achieved better performance compared with the systems submitted to SR'20 (Mille et al., 2020) where the model has to guess the functional words and morphological information. This high performance directly comes from the pretrained language model characteristics of BART. As for future work, we are planning to experiment with different graph linearization schemes. Also, we would like to use a multilingual language model such as mBART (Liu et al., 2020) on other languages as well as other language models. Finally, we would like to evaluate the incorporation of a copy mechanism (Gu et al., 2016) in generating long and complex tokens such as URLs.

Acknowledgements

The authors would like to thank the anonymous reviewers for their feedback on a previous version of this paper. This work was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 4585–4592, Reykjavik, Iceland, May. European Languages Resources Association (ELRA).

Wenchao Du and Alan W Black. 2019. Learning to order graph elements with application to multilingual surface realization. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 18–24.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale Seq2Seq models to multi-document inputs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 4186–4196, Hong Kong, China, November. Association for Computational Linguistics.

Farhood Farahnak, Laya Rafiee, Leila Kosseim, and Thomas Fevens. 2019. The Concordia NLG Surface Realizer at SRST 2019. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 63–67.

Y. Graham, Timothy Baldwin, A. Moffat, and J. Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Nat. Lang. Eng.*, 23:3–30.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August. Association for Computational Linguistics.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have your text and use it too! End-to-end neural data-to-text generation with semantic fidelity. *arXiv preprint arXiv:2004.06577*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Eduard Hovy, Gertjan van Noord, Guenter Neumann, and John Bateman. 1997. Language generation. *Survey of the State of the Art in Human Language Technology*, pages 131–146.

Mihir Kale. 2020. Text-to-text pre-training for data-to-text tasks. *arXiv preprint arXiv:2005.10433*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*, Ernest N. Morial Convention Center, New Orleans, May.

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 (ACL-2020)*, pages 1846–1852, Online, July. Association for Computational Linguistics.

Alessandro Mazzei and Valerio Basile. 2019. The dipinfounito realizer at srst'19: Learning to rank and deep morphology prediction for multilingual surface realization. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 81–87.

Simon Mille, Anja Belz, Bernd Bohnet, and Leo Wanner. 2018. Underspecified universal dependency structures as inputs for multilingual surface realisation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 199–209, Tilburg University, The Netherlands, November. 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), 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*, Hong Kong, China.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, New York, NY, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems (NIPS 2017)*, pages 5998–6008, Long Beach Convention Center, Long Beach.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations (ICLR 2018)*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pages 2692–2700. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Xiang Yu, Agnieszka Falenska, Marina Haid, Ngoc Thang Vu, and Jonas Kuhn. 2019. Imsurreal: Ims at the surface realization shared task 2019. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019) (EMNLP 2019)*, pages 50–58, Hong Kong, China.

Yao Zhou, Cong Liu, and Yan Pan. 2016. Modelling sentence pairs with tree-structured attentive encoder. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2912–2922, Osaka, Japan, December. The COLING 2016 Organizing Committee.