ADAPT at SR'20: How Preprocessing and Data Augmentation Help to Improve Surface Realization

Henry Elder
ADAPT Centre
Dublin City University
henry.elder@adaptcentre.ie

Abstract

In this paper, we describe the ADAPT submission to the Surface Realization Shared Task 2020. We present a neural-based system trained on the English Web Treebank and an augmented dataset, automatically created from existing text corpora.

1 Introduction

Surface realization is the final step of an NLG system (Reiter and Dale, 2000). The prior steps provide guidance on the content and structure of a sentence that is to be generated. The goal of this shared task is to generate sentences from structured data with high accuracy (Mille et al., 2020). Once this goal has been achieved, we believe neural-based surface realization systems could be incorporated into real-world NLG systems, such as task-oriented dialogue systems (Balakrishnan et al., 2019) and personalised marketing systems[1]

2 System Description

We made a submission to the Surface Realization Shared Task 2020, for the English language dataset: Universal Dependencies English Web Treebank (Silveira et al., 2014). We use a neural-based system; a sequence-to-sequence model trained on linearized trees. We submitted test outputs to both the open and closed tracks. For the open track we trained the same system on a large augmented dataset.

2.1 Data Preprocessing

Ablation analysis performed on our previous systems (Elder and Hokamp, 2018; Elder et al., 2020) showed that much of the performance comes from different preprocessing steps we apply to the original CoNLLU formatted data.

Figure 1 contains a formatted example of a linearized tree that is used as the input sequence when training the model. The output sequence used is the tokenized form of the original sentence. Below, we discuss the four key preprocessing features we use. More details can be found in the Python module used for preprocessing[2]; for each feature we point to the relevant lines of code in footnotes.

Depth First Linearizations To get the input sequence from a tree, we perform a depth first search of the tree[3]. This provides us with a linear sequence of tokens. Where a parent token has multiple child tokens, we choose randomly between the children. To ensure our system is robust to the order of the linearization, we obtain multiple random linearizations of each sentence to train the system with.

[1]For example `https://phrasee.co/` and `https://www.persado.com/`

[2]`https://github.com/Henry-E/surface-realization-shallow-task/blob/master/modules/create_source_and_target.py`

[3]`https://github.com/Henry-E/surface-realization-shallow-task/blob/master/modules/create_source_and_target.py\#L12-L36`

Proceedings of the Third Workshop on Multilingual Surface Realisation (MSR'20), pages 30–34
Barcelona, Spain (Online), December 12, 2020.

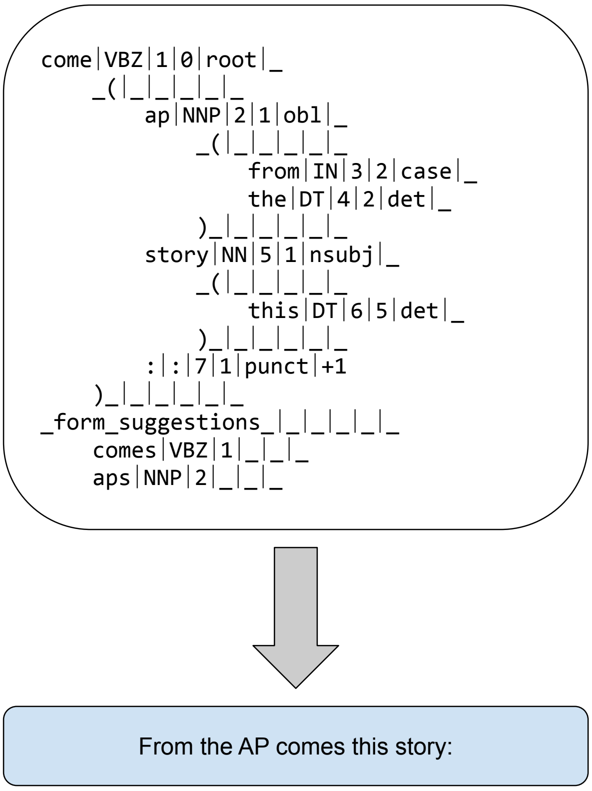

Figure 1: A formatted example of the linear input sequence for the sentence *From the AP comes this story:*. Token level features appear in the order: Lemma, XPOS, ID, Head, DepRel, Lin.

Scoping Brackets Similar to Konstas et al. (2017), we apply scoping brackets around child nodes. This provides further indication of the tree structure to the model, despite using a linear sequence as input[4].

Token Level Features We append a number of features to each token; XPOS, ID, HEAD, DepRel, Lin. This enables us to use factored sequence models (Sennrich and Haddow, 2016), which we will discuss in Section 2.3. To use this modelling feature a special pipe symbol, |, is required between each of the token's features[5].

Form Suggestions Finally, we address the problem of generating a token's form when only given its lemma. To do this, we provide the model with form suggestions[6]. Form suggestions are a list of possible forms that other lemmas, with the same XPOS tag, were observed to take. To obtain the form suggestions, we use the automatically parsed corpus, discussed in Section 2.2, to create a dictionary[7]. The dictionary is structured as such: each key is a concatenated lemma and XPOS tag, and the value is a list of possible forms observed in the automatically parsed corpus[8]. For example: {*"VBN_bootstrap": "bootstrapped"*}

2.2 Augmented Dataset

To augment the existing training data we create a dataset by parsing sentences from publicly available corpora. The two corpora we investigated are Wikitext 103 (Merity et al., 2017) and the CNN stories

[4] `https://github.com/Henry-E/surface-realization-shallow-task/blob/master/modules/create_source_and_target.py\#L23-L28`

[5] `https://github.com/Henry-E/surface-realization-shallow-task/blob/master/modules/create_source_and_target.py\#L79-L95`

[6] `https://github.com/Henry-E/surface-realization-shallow-task/blob/master/modules/create_source_and_target.py\#L101-L123`

[7] `https://github.com/Henry-E/surface-realization-shallow-task/blob/master/modules/get_form_suggestions.py`

[8] Dictionary: `https://github.com/Henry-E/surface-realization-shallow-task/blob/master/inflection_dicts/18th_october_tests/lemma_form_dict_sorted.json`

portion of the DeepMind Q&A dataset (Hermann et al., 2015).

Each corpus requires some cleaning and formatting, after which they can be sentence tokenized using CoreNLP (Manning et al., 2014). Sentences are filtered by length – min 5 tokens and max 50 – and for vocabulary overlap with the original training data – set to 80% of tokens in a sentence required to appear in the original vocabulary. These sentences are then parsed using the Stanford NLP UD parser (Qi et al., 2018). This leaves us with 2.4 million parsed sentences from the CNN stories corpus and 2.1 million from Wikitext. To convert a parse tree into the shared task format: word order information is removed by shuffling the IDs of the parse tree and tokens are lemmatised by removing the form column.

While it has been noted that the use of automatically created data is problematic in NLG tasks — WeatherGov (Liang et al., 2009) being the notable example — our data is created differently. The WeatherGov dataset is constructed by pairing a table with the output of a rule-based NLG system. This means any system trained on WeatherGov only re-learns the rules used to generate the text. Our approach is the reverse; we parse an existing, naturally occurring sentence, and, thus, the model must learn to reverse the parsing algorithm.

2.3 Model

The system is trained using a custom fork[9] of the OpenNMT-py framework (Klein et al., 2017), the only change made was to the beam search decoding code. The model used is a bidirectional recurrent neural network (BRNN) (Schuster and Paliwal, 1997) with long short term memory (LSTM) cells (Hochreiter and Schmidhuber, 1997). We trained two systems; one with the EWT dataset[10] and one with both the EWT dataset and our augmented dataset[11]. Hyperparameter details and replication instructions are available in our project's repository[12], in particular in the config directory. All hyperparameters stayed the same when training with the augmented dataset, except for vocabulary size and training time. Vocabulary size varies based on the datasets in use. It is determined by using any tokens which appears 10 times or more. When training on the EWT dataset, the vocabulary size is 2,193 tokens, training is done for 38 epochs and takes about 1 hour on two Nvidia 1080 Ti GPUs. For the combined EWT, Wikitext and CNN datasets the vocabulary size is 89,233, training time increases to around 2 days, and uses 60 random linearizations of the EWT dataset and 8 of the Wikitext and CNN datasets. The best performing checkpoint on the development set is chosen for testing.

Our system uses three non-standard modelling features, each of which performs a key function for the task:

Copy Attention Copy attention (Vinyals et al., 2015; See et al., 2017) gives models the ability to copy a token directly from the source sequence to the generated text, even if that token does not appear in the source vocabulary. Vocabularies are usually limited based on available data or computational constraints, so it's likely that at least some words the model sees during testing may not have been added to the vocabulary during training.

Factored Sequence Models Factored sequence models (Sennrich and Haddow, 2016) permit token level features to be used as part of training. The key idea is to create a separate embedding representation for each feature type, and to concatenate the embeddings to each token embedding to create a dense representation[13].

Restricted Beam Search In an attempt to reduce unnecessary errors during decoding, our beam search looks at the input sequence and restricts the available vocabulary to only tokens from the input, and tokens which have not yet appeared in the output sequence. This is similar to the approach used by King and White (2018).

[9] `https://github.com/Henry-E/OpenNMT-py`

[10] EWT Only Hyperparameters: `https://github.com/Henry-E/surface-realization-shallow-task/blob/master/configs/srst_2020/baseline_ewt.json\#L38-L62`

[11] Augmented Data Hyperparameters: `https://github.com/Henry-E/surface-realization-shallow-task/blob/master/configs/srst_2020/all_together.json\#L43-L67`

[12] `https://github.com/Henry-E/surface-realization-shallow-task`

[13] See Elder and Hokamp (2018) for more details

	BLEU	NIST	DIST
EWT	80.4	13.47	85.5
+ Augmented Corpora	87.5	13.81	90.35

Table 1: SR'20 EWT Test set results - Automated Evaluation metrics

3 Results

In this section we report our results on the shared task. An explanation of the evaluation methodology, as well as a comparison with other participants, can be found in the shared task description paper (Mille et al., 2020).

Table 1 contains automated evaluation metrics on the EWT test set. As in previous experiments (Elder et al., 2020), we find that the augmented dataset greatly improves the performance of our system.

System	Ave.	Ave. z	n	N
EWT + Augmented Corpora	75.7	0.426	797	913
HUMAN	**75.7**	**0.417**	**669**	**1,402**
EWT	72.5	0.32	830	953

Table 2: SR'20 Test set results - Human Evaluation: Readability

System	Ave.	Ave. z	n	N
EWT + Augmented Corpora	92.6	0.54	1,698	1,931
EWT	90.7	0.476	1,685	1,914

Table 3: SR'20 Test set results - Human Evaluation: Meaning Similarity

Table 2 contains human evaluation results for the readability metric. Rather surprisingly, the readability for our system with the augmented corpora is almost equivalent to the readability of the original human text. However, the readability metric only reflects how well written the annotators deemed a sentence to be. Readability scores don't take into account whether the generated sentence has managed to capture the meaning of the original sentence.

Table 3 contains human evaluation results for the meaning similarity metric. This metric describes how successful the system has been at generating sentences with the same meaning as the original sentence. Sentences generated by the augmented corpora are on average 92.6% similar in meaning to the original sentence. While this may seem like a strong result[14], ultimately we are aiming for 100% meaning similarity in order to have a system that is reliable enough to be used with real world NLG systems.

References

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained Decoding for Neural NLG from Compositional Representations in Task-Oriented Dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844, Stroudsburg, PA, USA. Association for Computational Linguistics.

Henry Elder and Chris Hokamp. 2018. Generating High-Quality Surface Realizations Using Data Augmentation and Factored Sequence Models. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 49–53, Stroudsburg, PA, USA. Association for Computational Linguistics.

Henry Elder, Robert Burke, Alexander O'Connor, and Jennifer Foster. 2020. Shape of Synth to Come: Why We Should Use Synthetic Data for English Surface Realization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7465–7471, Stroudsburg, PA, USA. Association for Computational Linguistics.

[14] The highest recorded meaning similarity on the same test set in last year's shared task was 86.6% (Mille et al., 2019)

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In C Cortes, N D Lawrence, D D Lee, M Sugiyama, and R Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

David King and Michael White. 2018. The OSU Realizer for SRST '18: Neural Sequence-to-Sequence Inflection and Incremental Locality-Based Linearization. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, number 2009, pages 39–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-Sequence Models for Parsing and Generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Stroudsburg, PA, USA, 4. Association for Computational Linguistics.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning Semantic Correspondences with Less Supervision. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, (August):91–99.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The {Stanford} {CoreNLP} Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer Sentinel Mixture Models. In *5th International Conference on Learning Representations, {ICLR} 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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)*, number Msr, pages 1–17, Stroudsburg, PA, USA. 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, 12. Association for Computational Linguistics.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. 2018. Universal Dependency Parsing from Scratch. In *Proceedings of the (CoNLL) 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium, 10. Association for Computational Linguistics.

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

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic Input Features Improve Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Stroudsburg, PA, USA. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A Gold Standard Dependency Corpus for {E}nglish. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation ({LREC}'14)*, pages 2897–2904, Reykjavik, Iceland, 5. European Language Resources Association (ELRA).

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Advances in Neural Information Processing Systems 28*, pages 2692–2700, 6.