The OSU-Facebook Realizer for SR '19: Seq2seq Inflection and Serialized Tree2Tree Linearization

Kartikeya Upasani,[1] **David L. King,**[2] **Jinfeng Rao,**[1]
Anusha Balakrishnan,[1] and **Michael White**[12]
[1]Facebook Assistant, Menlo Park, CA, USA
{kart,raojinfeng,anushabala,mwhite14850}@fb.com
[2]Department of Linguistics, The Ohio State University
The Ohio State University, Columbus, OH, USA
king.2138@osu.edu, mwhite@ling.osu.edu

Abstract

We describe our exploratory system for the shallow surface realization task, which combines morphological inflection using character sequence-to-sequence models with a baseline linearizer that implements a tree-to-tree model using sequence-to-sequence models on serialized trees. Results for morphological inflection were competitive across languages. Due to time constraints, we could only submit complete results (including linearization) for English. Preliminary linearization results were decent, with a small benefit from reranking to prefer valid output trees, but inadequate control over the words in the output led to poor quality on longer sentences.

1 Introduction

With our entry in the shallow surface realization shared task, we aimed to (1) implement an up-to-date morphological inflection model based on the approach of Faruqui et al. (2016) and Kann and Schütze (2016), and (2) conduct exploratory experiments with linearization using the constrained decoding approach of Balakrishnan et al. (2019) adapted to dependency trees.

Our system is a pipeline that begins by generating inflected wordforms from uninflected terminals in the tree using character seq2seq models. We then serialize these inflected syntactic trees as constituent trees by converting the relations to non-terminals. The serialized constituent trees are fed to seq2seq models (including models with copy and with tree-LSTM encoders), whose outputs also contain tokens marking the tree structure. We obtain n-best outputs for orderings and choose the highest confidence output sequence with a valid tree—i.e., one where the input and output trees are isomorphic up to sibling order—in order to obtain a projective linearization where possible, given that the vast majority of gold linearizations are projective.[1]

While we found that this validity checking step provided a small benefit, fully adapting the constrained decoding approach to dependency trees would have required adding a step to ensure that all and only the input words appeared in the output tree, and enforcing these constraints during beam search. Due to time constraints, however, we were only able to obtain preliminary linearization results for English without these word-level checks.

Development results for morphological inflection were competitive across languages as compared to previous implementations (King and White, 2018; Puzikov and Gurevych, 2018). With linearization, the preliminary results were decent, but showed substantial degradation for longer sentences where problems with lack of control over the output words became more severe.

In the rest of the paper, we describe our inflection and linearization components in more detail, along with our experimental results.

2 Inflection

Our pipeline begins by producing fully inflected word forms from the citation forms pro-

[1]To handle non-projective cases, the arc-lifting method of Bohnet et al. (2012) could be applied as a preprocessing step.

Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019), pages 68–74
Hong Kong, China, November 3rd, 2019.
https://doi.org/10.18653/v1/P17

vided in the UD input. In a sense, at this stage, the system has to be able to perform the wug test (Berko, 1958): having never seen a word before, we need to have the ability to produce the correct form for a given paradigm cell. We utilize sequence-to-sequence models (Bahdanau et al., 2014) in keeping with previous successful approaches (Kann and Schütze, 2016; Faruqui et al., 2016; King and White, 2018). Additionally, we reimplemented Kann and Schütze's 2016 approach in PyTorch.[2]

We also follow Kann and Schütze's approach by training our inflection model at the languages level and not at the level of individual paradigm cells as originally proposed by Faruqui et al. More formally, our LSTMs (Hochreiter and Schmidhuber, 1997) create an encoding, producing hidden state h_t which is dependent on the input x_t, the hidden state from the previous time step h_{t-1}, and nonlinear function f. c is the context of all previous time steps. Additionally, we set h_j to be the concatenated forward and backward encodings since we use bidirectional LSTMs.

$$h_t = f(x_t, h_{t-1}) \tag{1}$$

$$c = q(h_1, ..., h_{T_x}) \tag{2}$$

$$h_j = \left[\overrightarrow{h_j^T}, \overleftarrow{h_j^T}\right]^T \tag{3}$$

During inference (i.e. decoding), output y depends on the input sequence and previous inference steps. We also use the same attention as described by Bahdanau et al. and Kann and Schütze:

$$p(y|x) = \prod_{t=1}^{T_y} p(y_t|\{y_1, ..., y_t - 1\}, s_t, c_t) \tag{4}$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{5}$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{6}$$

$$e_{ij} = a(s_{i-1}, hj) \tag{7}$$

As seen in Table 1, uncased results are almost always higher than cased. This should not surprise us as, operating on the word-internal level, any sequence-to-sequence model would have no access to syntagmatic information outside of how UDs encode that information in the morphosynctactic feature sets. Also Arabic, Hindi, and Japanese do not have cased orthography and therefore have no difference in their case/uncased accuracies.

As for feature sets, we include the same set as described by King and White. In addition to using the morphosyntactic features provided by the UD schema, we also used the POS tag and dependency name as input to the system. Differing from previous shared tasks (Cotterell et al., 2016, 2017), we do not alter the token frequencies. In traditional SIGMORPHON inflection tasks, each system only sees a word form once per epoch. We found that this causes the system to miss irregulars. Since irregular forms tend to occur with higher frequency, allowing the system to see more examples during each epoch increased performance on irregular forms. We also found that adding a rule for English specifically designed to account for the "to be" paradigm raises accuracy for English another 0.6% to 98.5%.

Finally, for Korean and Chinese, we simply write rule sets for their morphology. The Korean dataset exclusively uses concatenation. The input forms list items and their corresponding affixes, in order, and simply removing the morpheme boundary token (a "+") yielded 100% accuracy. For Chinese, the plural marker "们" (*men*) only ever occurred with "人" (*rén*, "person"), "我" (*wǒ*, "I"), "🄵" (*tā*, "it" [animals]), "它" (*tā*, "it" [inanimate]), "她" (*tā*, "she"), and "他" (*tā*, "he"). Writing a rule that adds "们" when any of the character co-occur with the `Num=Plur` feature also gives us 100% accuracy for Chinese.

3 Linearization

To help assess the potential of using tree-to-tree models with constrained decoding (Balakrishnan et al., 2019) for linearization and guide future work in this direction, we conducted exploratory experiments using off-the-shelf sequence-to-sequence models where the input and output trees are represented as sequences using non-terminal tokens corresponding to dependency relations. In these serialized trees, each non-terminal token is followed by the inflected form, its dependents,

[2]Freely available here: https://github.com/davidlking/med-pytorch

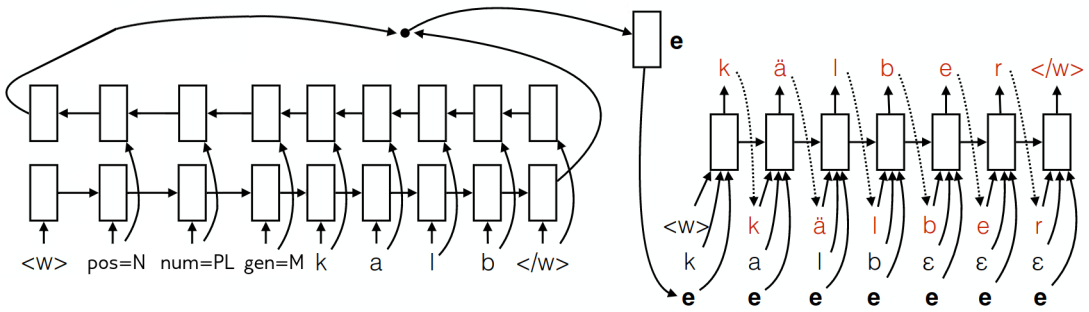

Figure 1: A graphical representation of the architecture originally introduced by Faruqui et al. (2016) and adapted by Kann and Schütze (2016). A bidirectional LSTM creates an encoding of the input wordform and supplied features. That encoding is subsequently fed to the decoder LSTM along with the original input wordform.

	Language										
Model	ar	en	es	fr	hi	id	ja	ko	pt	ru	zh
Cased	92.2	91.1	91.3	89.2	97.3	86.4	99.6	N/A	87.1	90.0	N/A
Uncased	92.2	97.9	93.5	95.3	97.3	98.9	99.6	N/A	91.4	96.9	N/A

Table 1: Morphological inflection results on the development set. Although we only submitted results for the English due to time constraints, we did train inflection models for each language.

Input: **[root** behind **[advmod** badly **]** **[advmod** now **]** [cop are **]** **[nsubj** we **]** **[obl** matter **[case** in **]** **[det** this **]** **]** **[punct** . **]** **]**

Target: **[root** **[nsubj** we **]** **[cop** are **]** **[advmod** badly **]** behind **[advmod** now **]** **[obl** **[case** in **]** **[det** this **]** matter **]** **[punct** . **]** **]**

Figure 2: Example of serialized tree representation used for linearization.

and finally a closing-bracket indicating the end of the non-terminal's span, as exemplified in Figure 2 shows an example of serialized inputs and outputs.

We experimented with three different variants of sequence-to-sequence models:

Seq2Seq: Simple encoder-decoder model with attention (Bahdanau et al., 2014). Both the encoder and decoder are LSTMs.

Tree2Seq: Similar to `Seq2Seq`, but we use a variant of the N-ary tree-LSTM (Tai et al., 2015) as the encoder, as described in Rao et al. (2019), thereby potentially taking better advantage of the input tree structure.

Seq2Seq-Copy: `Seq2Seq` model with a pointer-generator mechanism (See et al., 2017) for copying tokens from input. The decoder can choose to either generate a word from the vocabulary or copy an input token instead. We did not have an off-the-shelf implementation for a Tree2Seq-Copy model, though our experiments suggest it would be worth developing one.

Additionally, we also experimented with constrained decoding (Balakrishnan et al., 2019) with each of the above model. Using this method, in each step of beam search, we check for and remove candidates whose tree structures deviate from that of the input tree. The constraints include ensuring that a parent node only accepts valid children, and that all its children have been generated before it can accept a closing bracket, thereby helping to ensure a projective realization. However, as noted in the introduction, we did not have time to extend the constraints to ensure that all and only the input words appeared in the output, so we did not expect this method to work as well as we would have liked. As such, we also experimented with reranking an n-best list to select the highest-scoring output with a valid tree (i.e., one that matches the tree of the input, up to sibling ordering).

4 Results

We picked the approach that gave the best performance on dev set. We combined samples of all English train sets, training on all sets together gave better dev BLEU scores than training individually. Table 2 shows a comparison of the different models that we tried.

Model	en_gum-ud	en_partut-ud
Seq2Seq	0.180	0.163
Tree2Seq	0.585	0.275
Seq2Seq-Copy	0.870	0.902

Table 2: Dev set BLEU scores (calculated along with non-terminals), using gold inflected forms

In the table, the BLEU scores are calculated with the non-terminals included in both input and output sequences, inflating them somewhat relative to regular BLEU scores. Gold inflected forms were also used.

Table 3 compares the constrained and unconstrained versions of the `Seq2Seq-Copy` (again with non-terminals in the output and gold inflected forms). Since we did not have time to implement word-level constraints, the results seem to be mixed. In the end, we chose the constrained model on datasets where dev BLEU was higher than its unconstrained counterpart. Table 4 shows the gains obtained by doing validity reranking (again with gold inflected forms); here the scores shown are calculated without non-terminals.

Given our time constraints, we only submitted English results for evaluation. Although we generated inflected forms for all languages in the T1 task, we could only obtain linearization results for English. Our results are decent (with the exception of the en_partut-ud-test dataset), suggesting that the approach may represent a viable starting point for future work. In particular, in the human evaluation results for English in the shared task overview paper (Mille et al., 2019), our system was ranked in the middle group of systems for meaning preservation and in the large group of systems tied for third–twelfth place in readability. Consistent with the human evaluation, the automatic scores for our system (Table 5) were also in the middle of the pack. Note that the test scores are lower than the dev scores at least in part because only the former are calculated with generated inflected forms.

5 Discussion

Regarding the en_partut-ud-test dataset, our preliminary error analysis seems to indicate that the inflection model overfit the dev set. Although the model outputs relatively sane er-

Model	en_ewt-ud	en_gum-ud	en_partut-ud
Seq2Seq-Copy Unconstr	0.8239	0.8731	0.8984
Seq2Seq-Copy Constr	0.8499	0.8337	0.8847

Table 3: Preliminary BLEU scores for constrained decoding (calculated along with non-terminals), using gold inflected forms

Dataset	w/o reranking	w/ reranking
en_ewt	0.8328	0.8405
en_gum	0.8294	0.8289
en_lines	0.7655	0.7778
en_partut	0.7891	0.7909

Table 4: BLEU scores on dev sets before and after reranking, using gold inflected forms

Test set	BLEU	NIST	DIST
en_ewt-ud-test	62.38	11.29	77.93
en_gum-ud-test	49.91	8.5	66.88
en_lines-ud-test	54.56	9.89	71.07
en_partut-ud-test	7.37	3.21	54.27
en_pud-ud-test	67.91	11.74	78.12
en_ewt-Pred-HIT-edit	60.58	10.96	74.64
en_pud-Pred-LATTICE	66.18	11.7	76.8

Table 5: Test set results for English from the organizers

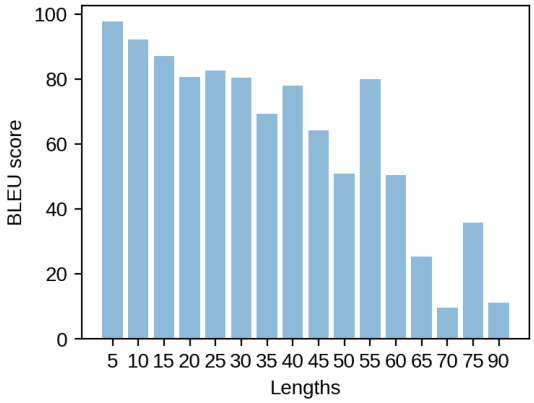

Figure 3: BLEU score plotted against gold sequence lengths for en-gum-ud dev set.

rors with the other test sets, errors with this particular set are much noisier. For example, in another file the model emits "multichart" as "multichartart". This kind of error is extremely consistent with errors regarding the attention mechanism. In fact, Faruqui et al. explicitly feed the lemma into their decoder for this very reason. That said, errors from the en_partut-ud-test file are not as clear (e.g. "copyright" → "Sropopyright").

Turning to linearization, `Seq2Seq-Copy` does much better than the other models. We believe this is due to the architectural prior of copying words from the input, as nearly all output words are present in the input (modulo words whose inflected forms are sensitive to adjacent words). Figure 3 shows that BLEU scores significantly decrease as sequence length increases. Figure 4 shows that the number of extra or missing words increases with lengths, which could explain the drop in BLEU. Such mistakes could perhaps have been avoided by adding word-level constraints to constrained decoding. Other errors are due to picking

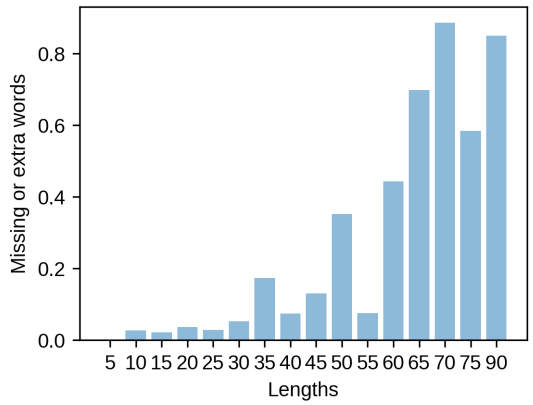

Figure 4: Extra or missing words normalized by length, plotted against gold sequence lengths for en-gum-ud dev set.

the correct words but in the incorrect order. Out of 366 mismatches on the en-gum dev set, 193 (53%) are cases of mismatched word order with the correct words.

Figure 5 shows examples of linearization model predictions. In 1, the model misses the word "summer" and repeats "olympic" instead. This can potentially be alleviated by constraining the generation of a word based on the number of times it appears in the input. In 2, the model picks the right set of words but in an order that is different from the gold order. In 3, the model fails by stuttering, i.e. it repeats the same phrase again and again.

6 Conclusions and Future Work

Our exploratory experiments show that combining a morphological inflection with a baseline linearizer achieves decent results. Our pipeline for the shallow surface realization shared task first produces inflected wordforms from lemmas using a character level sequence-to-sequence model. We then use those forms in serialized trees as input to a tree-to-tree model, which is also implemented using a sequence-to-sequence architecture, yielding serialized trees as output. This allows outputs to be filtered for validity in most cases, enforcing projective outputs. Due to time limitations we could only submit fully linearized results for English, and we were not able to implement word-level constraints, so we consider these preliminary baseline results. Given our error analysis, in future work it may be fruitful to update the attention mechanism in the inflection model (Aharoni and Goldberg, 2017), and to use a tree encoder + copy mechanism in the linearizer together with word-level constraints in decoding.

References

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2004–2015.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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, Florence, Italy. Association for Computational Linguistics.

Jean Berko. 1958. The child's learning of english morphology. *Word*, 14(2-3):150–177.

Bernd Bohnet, Anders Björkelund, Jonas Kuhn, Wolfgang Seeker, and Sina Zarriess. 2012. Generating non-projective word order in statistical linearization. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 928–939, Jeju Island, Korea. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 Shared Task: Universal morphological reinflection in 52 languages. *CoRR*, abs/1706.09031.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 Shared Task—Morphological Reinflection. In *Proceedings of the 2016 Meeting of SIGMORPHON*, Berlin, Germany. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. In *Proc. of NAACL*.

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

Katharina Kann and Hinrich Schütze. 2016. MED: The LMU system for the SIGMORPHON 2016 shared task on morphological reinflection. *ACL 2016*, page 62.

David L King and Michael White. 2018. The osu realizer for srst' 18: Neural sequence-to-sequence inflection and incremental locality-based linearization. *ACL 2018*, page 39.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong, China.

1. **Gold**: athens hosted the 2004 summer olympic games .
   Pred: athens hosted the olympic olympic 2004 games .

2. **Gold**: and it has , because it is spring and inside the ground something is stirring .
   Pred: and it has because it is spring , and something is stirring inside the ground .

3. **Gold**: ok well i have a crush on 2 guys but unforchunitly it is almost valentine s day afnd i just broke up with my and i have dated one of the guys i like and one guy lives in my naborhood guy i need your help i am a girl but i need a guy s help what shoul i do ?
   Pred: well i have a crush on 2 guy and i have almost dated one of the guys i like and i have almost dated one of the guys i like and i have almost dated one of the guys i like and i have almost dated one of the guys i like and i have almost dated one of the guys i like ?

Figure 5: Examples of linearization predictions with BLEU score < 0.6.

Yevgeniy Puzikov and Iryna Gurevych. 2018. BinLin: A simple method of dependency tree linearization. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 13–28, Melbourne, Australia. Association for Computational Linguistics.

Jinfeng Rao, Kartikeya Upasani, Anusha Balakrishnan, Michael White, Anuj Kumar, and Rajen Subba. 2019. A tree-to-sequence model for neural nlg in task-oriented dialog. In *Proceedings of the 12th International Conference on Natural Language Generation (to appear)*.

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, Vancouver, Canada. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.