

# Surface Realization Shared Task 2019 (SR'19): The Tilburg University Approach

Thiago Castro Ferreira<sup>1,2</sup> and Chris van der Lee<sup>1</sup> and Emiel Krahmer<sup>1</sup>

<sup>1</sup>Tilburg center for Cognition and Communication (TiCC), Tilburg University, The Netherlands

<sup>2</sup>Federal University of Minas Gerais (UFMG)

{tcastrof, e.j.krahmer}@tilburguniversity.edu

## Abstract

This study describes the approach developed by the Tilburg University team to the shallow track of the Multilingual Surface Realization Shared Task 2019 (SR'19) (Mille et al., 2019). Based on Ferreira et al. (2017) and on our 2018 submission Ferreira et al. (2018), the approach generates texts by first preprocessing an input dependency tree into an ordered linearized string, which is then realized using a rule-based and a statistical machine translation (SMT) model. This year our submission is able to realize texts in the 11 languages proposed for the task, different from our last year submission, which covered only 6 Indo-European languages. The model is publicly available<sup>1</sup>.

## 1 Introduction

This study presents the approach developed by the Tilburg University team for the shallow track of the Multilingual Surface Realization Shared Task 2019 (SR'19) (Mille et al., 2019). Given a lemmatized dependency tree without word order information, the goal of this task consists of linearizing the lemmas in the correct order and realizing them as a surface string with the proper morphological form.

Our approach is similar to our submission for the 2018 version of the shared-task (Ferreira et al., 2018). It is based on the surface realization approach described in Ferreira et al. (2017), where a semantic graph structure is first preprocessed into a preordered linearized form, which is subsequently converted into text using a Statistical Machine Translation (SMT) model implemented in

<sup>1</sup><https://github.com/ThiagoCF05/Dep2Text>

Moses (Koehn et al., 2007). The difference is that, instead of a semantic structure, our approach preprocesses the lemmas of the dependency tree into an ordered linearized version, which is then converted into text using rules and an SMT model.

Different from our last submission where our approach covered only some of the proposed languages (6 out of 10), this year it is able to generate text in all of the 11 languages proposed in the shared-task: Arabic, Chinese, English, French, Hindi, Indonesian, Japanese, Korean, Portuguese, Russian and Spanish. For these languages, parallel datasets were provided with alignment information between source and target sides.

Regarding the languages covered in the previous version of the shared-task, our submission introduced promising results for English, French, Portuguese and Spanish, with BLEU scores higher than 40. For the newly covered languages, results appear promising for realizing Hindi and Indonesian output, with BLEU scores higher than 50. However, the approach appeared to work poorly for Arabic and Russian, and had problems to generate texts in the Asian languages Chinese, Japanese and Korean.

In the remainder of this paper, we better describe our method: Section 2 describes the general approach, Section 3 describes the results and discussion of our approach and Section 4 concludes the study, also describing future work which can be done to improve the model.

## 2 Model

Following our submission of last year (Ferreira et al., 2018), our model is based on the NLG approach introduced in Ferreira et al. (2017), where a semantic graph structure is first preprocessed into a preordered linearized form, which is then converted into its textual counterpart using an SMT

model implemented with Moses. However for this task, instead of a semantic structure, our approach takes as input a lemmatized dependency tree, which is linearized and converted into its final version by a rule-based and an SMT model. In the next sections, we explain the linearization and realization phases in more detail.

## 2.1 Linearization

This method aims to linearize a dependency tree input without punctuation nodes into an ordering string format. Our approach is similar to the 2-step classifier introduced in Ferreira et al. (2017) and is depicted in Algorithm 1.

The approach starts by deciding which first-order child nodes are most likely to be before and after its head node (lines 1-13). It uses a maximum entropy classifier  $\phi_1$ , trained for each language based on the relevant aligned training set. As features, this classifier uses the lemmas as well as the dependency and part-of-speech tags of the head and child nodes.

Once the nodes are split into a group of nodes before and another group of nodes after their heads, each one of these groups is ordered with an algorithm similar to the MergeSort one (lines 14-24 and function *SORT*). To decide the order of two child nodes of a same group, we use a second maximum entropy classifier  $\phi_2$ , also trained for each language based on the corresponding aligned training set. As features (line 44), it uses the lemmas as well as the dependency and part-of-speech tags of the head and the two child nodes involved in each comparison.

## 2.2 Realization

Once the dependency trees are linearized, two methods were used to surface realize the lemmas: a rule-based and a statistical machine translation (SMT) model.

**Rule-based** For all the 11 covered languages, this approach uses a lexicon created based on the aligned information extracted from the datasets. Given a lemma and its features, our approach looks for the most frequent morphological form in the lexicon.

**SMT** For 4 languages (English, French, Portuguese and Spanish), after linearizing the dependency tree and realizing the lemmas using a rule-based strategy, we trained a phrase-based machine translation to convert this representation into the

---

### Algorithm 1 Linearization method

---

```

Require: depTree
1: function LINEAR(root, orderId)
2:   before  $\leftarrow \emptyset$ 
3:   after  $\leftarrow \emptyset$ 
4:   edges  $\leftarrow$  getEdges(depTree, root)
5:   for all edge  $\in$  edges do
6:     node  $\leftarrow$  edge.node
7:     features1  $\leftarrow$  f1(depTree, root, node)
8:     if  $\phi_1$ (features1) == before then
9:       before  $\leftarrow$  before  $\cup$  node
10:    else
11:      after  $\leftarrow$  after  $\cup$  node
12:    end if
13:  end for
14:  before  $\leftarrow$  SORT(before)
15:  for all node  $\in$  before do
16:    orderId  $\leftarrow$  LINEAR(node, orderId)
17:  end for
18:  root.orderId  $\leftarrow$  orderId
19:  orderId  $\leftarrow$  orderId + 1
20:  after  $\leftarrow$  SORT(after)
21:  for all node  $\in$  after do
22:    orderId  $\leftarrow$  LINEAR(node, orderId)
23:  end for
24:  return orderId
25: end function
26:
27: function SORT(nodes)
28:   if |nodes| < 2 then
29:     return nodes
30:   end if
31:   half  $\leftarrow$  |nodes|/2
32:   end  $\leftarrow$  |nodes|
33:   nodes1  $\leftarrow$  SORT(nodes[0, half])
34:   nodes2  $\leftarrow$  SORT(nodes[half, end])
35:   ordNodes  $\leftarrow \emptyset$ 
36:   i1, i2  $\leftarrow$  0, 0
37:   while i1 < |nodes1| or i2 < |nodes2| do
38:     if |nodes1| = 0 then
39:       ordNodes  $\leftarrow$  ordNodes  $\cup$  POP(nodes2)
40:       i2  $\leftarrow$  i2 + 1
41:     else if |nodes2| = 0 then
42:       ordNodes  $\leftarrow$  ordNodes  $\cup$  POP(nodes1)
43:       i1  $\leftarrow$  i1 + 1
44:     else
45:       node1  $\leftarrow$  POP(nodes1)
46:       node2  $\leftarrow$  POP(nodes2)
47:       features2  $\leftarrow$  f2(depTree, node1, node2)
48:       if  $\phi_2$ (features2) = before then
49:         ordNodes  $\leftarrow$  ordNodes  $\cup$  node1
50:         i1  $\leftarrow$  i1 + 1
51:       else
52:         ordNodes  $\leftarrow$  ordNodes  $\cup$  node2
53:         i2  $\leftarrow$  i2 + 1
54:       end if
55:     end if
56:   end while
57:   return ordNodes
58: end function
59:
60: LINEAR(depTree.root, 0)

```

---

Language	Model	In/Out-Dom.	File	BLEU	DIST	NIST
Arabic	Rule	in-domain	ar_padt	19.01	6.40	53.82
Chinese	Rule	in-domain	zh_gsd	0.16	0.05	58.78
English	SMT	in-domain	en_ewt	56.28	10.88	74.39
		in-domain	en_gum	57.37	10.41	70.54
		in-domain	en_lines	53.78	10.09	67.88
		in-domain	en_partut	62.08	9.19	67.08
		out-domain	en_pud	60.04	11.04	71.75
		predicted	en_ewt	56.25	10.89	73.33
French	SMT	predicted	en_pud	55.67	10.83	67.99
		in-domain	fr_gsd	45.42	9.20	63.46
		in-domain	fr_partut	60.04	8.55	72.35
Hindi	Rule	in-domain	fr_sequoia	50.14	9.47	66.37
		in-domain	hi_hdtb	61.09	12.26	65.73
Indonesian	Rule	predicted	hi_hdtb	61.90	12.37	66.21
		in-domain	id_gsd	52.55	10.51	71.77
Japanese	Rule	in-domain	ja_gsd	0.14	0.01	55.37
		out-domain	ja_pud	0.08	0.01	52.86
Korean	Rule	in-domain	ko_gsd	0.00	0.00	31.35
		in-domain	ko_kaist	0.00	0.00	31.50
		predicted	ko_kaist	0.00	0.00	34.07
Portuguese	SMT	in-domain	pt_bosque	46.31	9.37	63.79
		in-domain	pt_gsd	35.43	9.00	59.89
		predicted	pt_bosque	47.85	9.60	64.76
Russian	Rule	in-domain	ru_gsd	6.65	4.50	50.58
		in-domain	ru_syntagrus	29.59	10.07	57.28
		out-domain	ru_pud	15.54	6.41	59.11
Spanish	SMT	in-domain	es_ancora	54.64	11.73	63.27
		in-domain	es_gsd	49.00	9.86	62.70
		predicted	es_ancora	55.04	11.73	63.5

Table 1: BLEU, DIST and NIST scores of our approach in the original (non-tokenized) test sets.

final realized text. The SMT model was built using the Moses toolkit (Koehn et al., 2007).

The settings were copied from the Statistical MT system introduced in Ferreira et al. (2017). At training time, we extract and score phrases up to the size of nine tokens. As feature functions, we used direct and inverse phrase translation probabilities and lexical weighting, as well as word, unknown word and phrase penalties. These feature functions were trained using alignments from the training set obtained by MGIZA (Gao and Vogel, 2008). Model weights were tuned on the development data using 60-batch MIRA (Cherry and Foster, 2012) with BLEU as the evaluation metric. A distortion limit of 6 was used for the reordering models. We used two lexicalized reordering models: a phrase-level (phrase-msd-bidirectional-fe) (Koehn et al., 2005) and a hierarchical-level one (hier-mslr-bidirectional-fe) (Galley and Manning, 2008). At decoding time, we used a stack size of 1000. To rerank the candidate texts, we used a 5-gram language model trained on the EuroParl corpus (Koehn, 2005) using KenLM (Heafield, 2011).

### 3 Results and Discussion

Concerning the languages covered in the previous version of the shared-task, our approach introduced promising results for English, French, Portuguese and Spanish, with BLEU scores higher than 40. For the newly covered languages, results were promising for the realization of Hindi and Indonesian texts, with BLEU scores higher than 50. On the other hand, our approach obtained low results for Arabic and Russian, and had problems to generate texts in the Asian languages Chinese, Japanese and Korean. For Chinese and Japanese, the problem arose from the fact we did not manage the tokenization/detokenization process well, which had a drastic negative influence on the final results.

### 4 Conclusion

This study described a shallow surface realizer for the 11 target languages in the Surface Realization Shared Task 2019 (SR’19). In future work, we aim to fix the problems for the Asian languages Chinese, Japanese and Korean. Specifically, for Chinese and Japanese, we require a proper method to tokenize/detokenize the results produced by our

approach. Moreover, we aim to design the task based on novel pipeline architectures for Natural Language Generation (Ferreira et al., 2019).

## Acknowledgments

This work is part of the research program “Discussion Thread Summarization for Mobile Devices” (DISCOSUMO) which is financed by the Netherlands Organization for Scientific Research (NWO).

## References

- Colin Cherry and George Foster. 2012. [Batch tuning strategies for statistical machine translation](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT’12*, pages 427–436, Montreal, Canada. Association for Computational Linguistics.
- Thiago Castro Ferreira, Iacer Calixto, Sander Wubben, and Emiel Kraemer. 2017. [Linguistic realisation as machine translation: Comparing different mt models for amr-to-text generation](#). In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 1–10. Association for Computational Linguistics.
- Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Kraemer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. *arXiv preprint arXiv:1908.09022*.
- Thiago Castro Ferreira, Sander Wubben, and Emiel Kraemer. 2018. [Surface realization shared task 2018 \(SR18\): The Tilburg university approach](#). In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 35–38, Melbourne, Australia. Association for Computational Linguistics.
- Michel Galley and Christopher D. Manning. 2008. [A simple and effective hierarchical phrase reordering model](#). In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, EMNLP’08*, pages 848–856, Honolulu, Hawaii. Association for Computational Linguistics.
- Qin Gao and Stephan Vogel. 2008. [Parallel implementations of word alignment tool](#). In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing, SETQA-NLP’08*, pages 49–57, Columbus, Ohio. Association for Computational Linguistics.
- Kenneth Heafield. 2011. [Kenlm: Faster and smaller language model queries](#). In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT’11*, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.
- Philipp Koehn, Amittai Axelrod, Alexandra Birch, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *International Workshop on Spoken Language Translation*.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. [Moses: Open source toolkit for statistical machine translation](#). In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL’07*, pages 177–180, Prague, Czech Republic. 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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong, China.