Graph-to-Graph Transformer for Transition-based Dependency Parsing

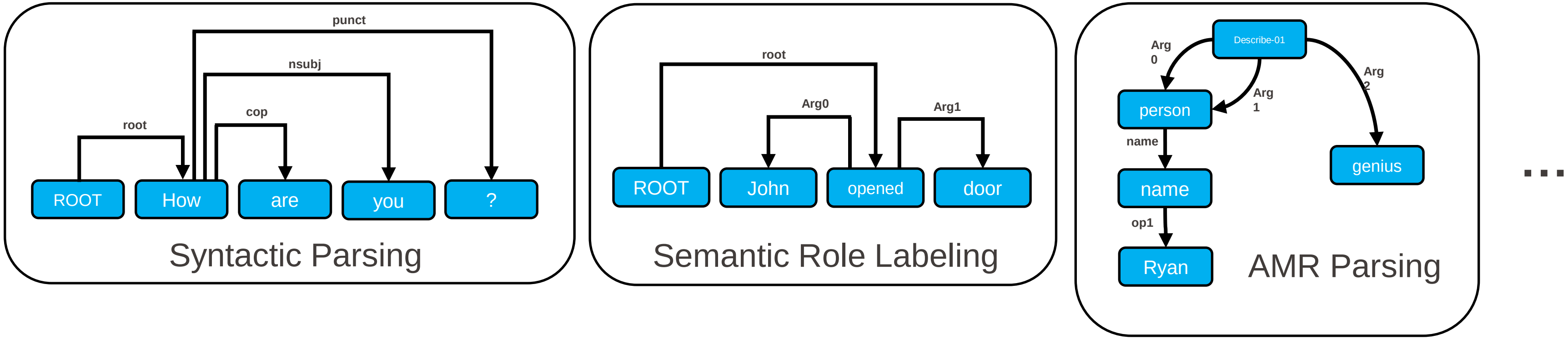
Alireza Mohammadshahi[□◊], James Henderson[◊]

◊ **IDIAP Research Institute** □ **École Polytechnique Fédérale de Lausanne-EPFL**

Email: {alireza.mohammadshahi,james.henderson}@idiap.ch

Motivation

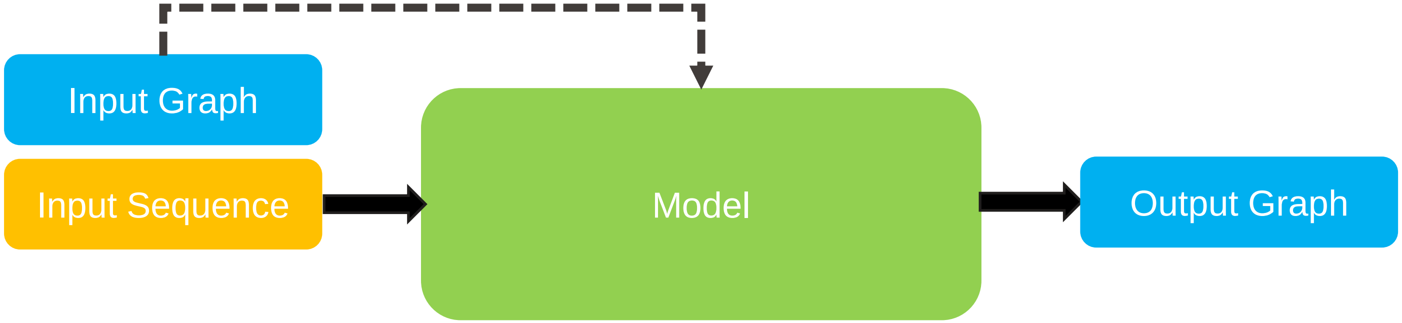
- Several NLP tasks interact with different graphs:



- Or, using graph structure as additional input e.g. NLI, MT

Motivation

- We need a deep learning architecture which can input and output any kind of graph:



Our Proposal

We propose **Graph-to-Graph Transformer (G2GTr)** architecture:

- Define a general encoder that encodes **both sequence and graph**
- Output a graph for the downstream task
- Works with pre-trained attention-based Models, e.g. **BERT**
- Achieve **state-of-the-art** results in transition-based dependency parsing

Graph-to-Graph Transformer

We have input sequence $\boldsymbol{X}$, Transformer finds Output representation $\boldsymbol{Z}$:

$$z_i = \sum_j \alpha_{ij}(x_j \boldsymbol{W}^V)$$

Attention weights are calculated as:

$$\alpha_{ij} = \frac{e_{ij}}{\sum_{k=1}^{n} e_{ik}} \quad , \qquad e_{ij} = \frac{(x_i \boldsymbol{W}^Q)(x_i \boldsymbol{W}^K)}{\sqrt{d}}$$

- $\boldsymbol{W}^V, \boldsymbol{W}^Q, \boldsymbol{W}^K$ are value, query, and key matrices.

Graph-to-Graph Transformer

To input a graph, we modify equations of Transformer:

$$z_i = \sum_j \alpha_{ij}(x_j \boldsymbol{W^V} + p_{ij}\boldsymbol{W_2^L})$$

Attention weights are calculated as:

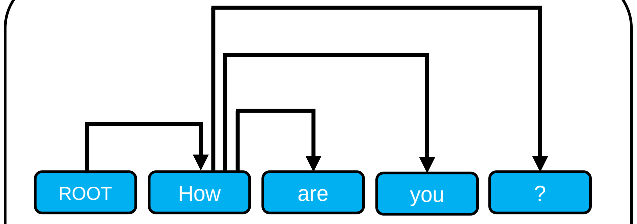
$$\alpha_{ij} = \frac{e_{ij}}{\sum_{k=1}^{n} e_{ik}} \quad , \qquad e_{ij} = \frac{(x_i \boldsymbol{W^Q})(x_i \boldsymbol{W^K} + p_{ij}\boldsymbol{W_1^L})}{\sqrt{d}}$$

- $\boldsymbol{p_{ij}}$ is the graph relation between token x_i and x_j.

Graph-to-Graph Transformer

- Attention value representation can contain both token-level and graph-level information.

- Matrix $P \in R^n \times R^n$ can be constructed with any input graph (n is sequence length).



Unlabelled Syntactic Graph

1	*head – dep*
2	*dep – head*
0	*None*

	ROOT	How	are	you	?
ROOT	0	1	0	0	0
How	2	0	1	1	1
are	0	2	0	0	0
you	0	2	0	0	0
?	0	2	0	0	0

Graph-to-Graph Transformer

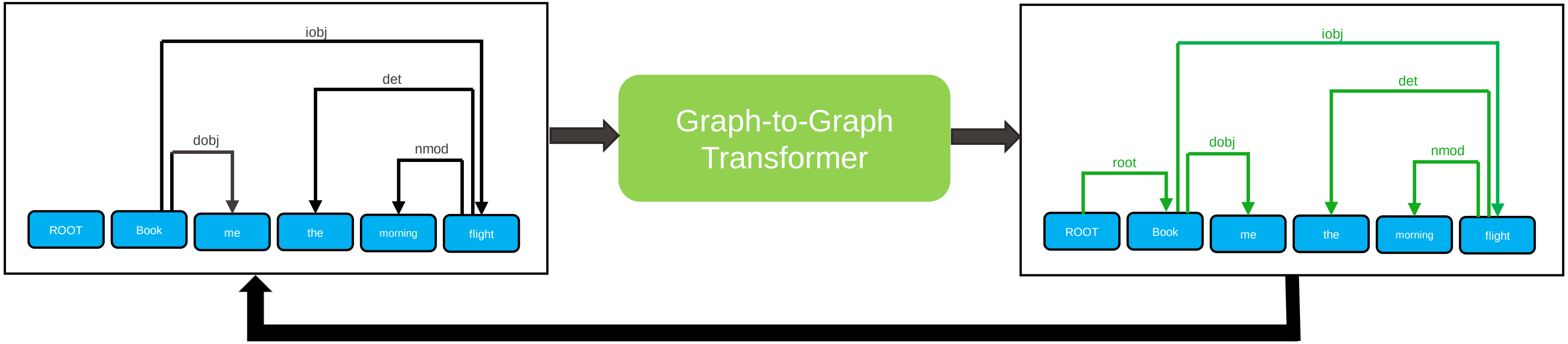
- Can be applied to any **NLP tasks** which require to **input a graph** or **produce a graph** over the same nodes.

- In this paper, we apply it to **transition-based dependency parsing**.

- In transition-based parsing, the model predicts a **new relation** based on the **parser state** (stack+buffer).

Our Transition-based Model

- Iteratively builds the dependency graph in an auto-regressive manner:



Our Transition-based Model

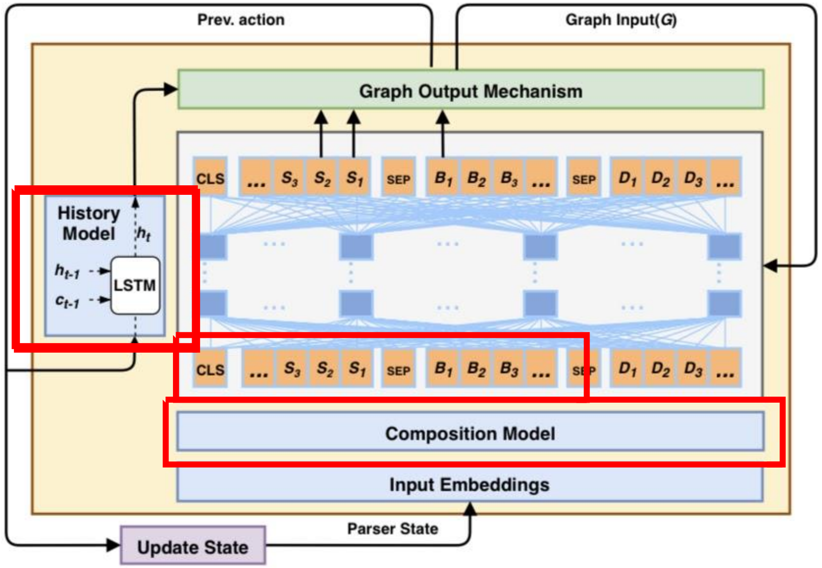
Our novel attention-based parsers:

- State Transformer (StateTr)
- Sentence Transformer (SentTr)

State Transformer

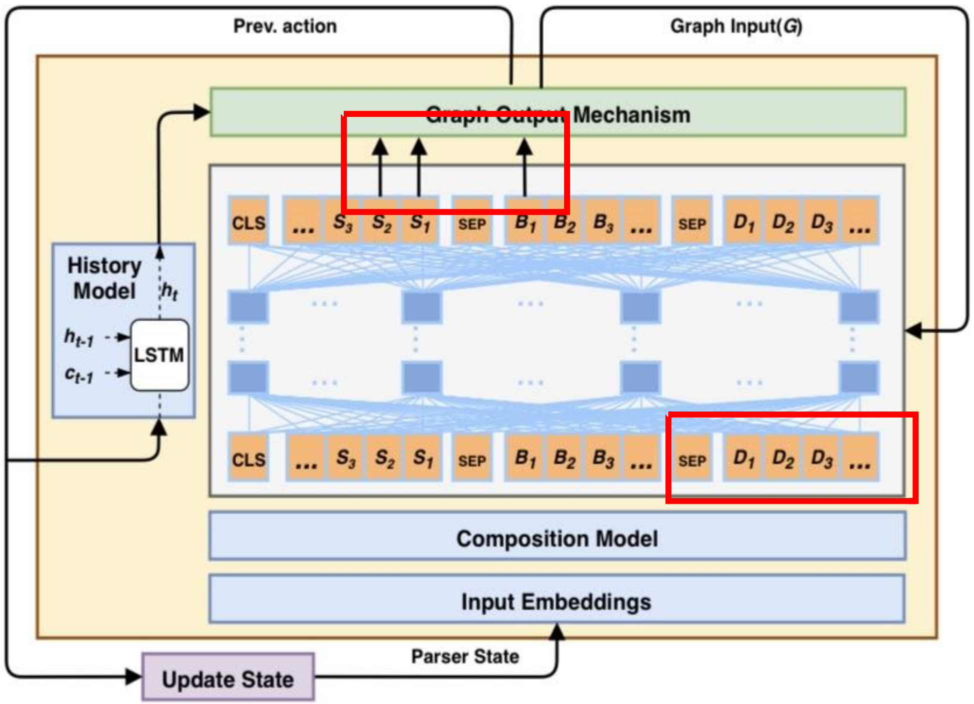
- Directly inputting the current state of the parser to Transformer.
- Contains additional History and Composition models.

- History Model:
  - Keeps track of previous predictions
- Composition Model:
  - An alternative to encoding partial graphs
  - Inspired by (Dyer et al,2015)
  - More details in the paper



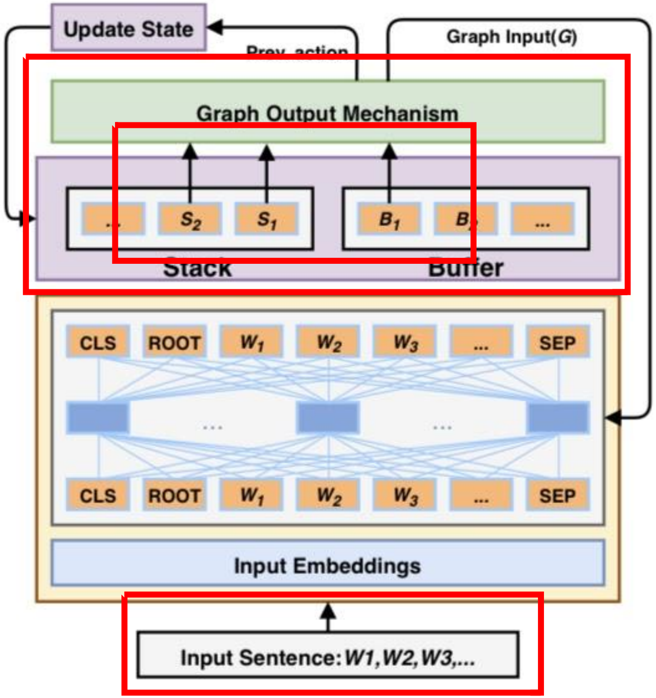
StateTr+G2GTr

- Graph Input Mechanism:
  - Dependencies between words
  - Add a third part (D) keeps track of words that have been deleted from the stack.
- Graph Output Mechanism:
  - Action prediction
  - Label prediction



SentTr+G2GTr

- Inputting initial sentence to G2GTr, then predicting based on parser state
- Graph input mechanism:
  - Dependencies between tokens
- Graph output mechanism:
  - Action prediction
  - Label prediction



WSJ Results

- G2GTr integration:
  - Without BERT pre-training
  - With BERT pre-training
  - Comparison with StackLSTM
- Replacement of Composition function
- Graph output mechanism
- State-of-the-art results on WSJ Penn Treebank

Model	Dev Set		Test Set	
	UAS	LAS	UAS	LAS
(Dyer et al, 2015)			93.1	90.9
(Weiss et al,2015)			94.26	91.42
(Cross and Huang, 2016)			93.42	91.36`
(Ballesteros et al,2016)			93.56	92.41
(Andor et al, 2016)			**94.61**	**92.79**
(Kiperwasser,2016)			93.90	91.9
(Yang et al,2017)			94.18	92.26
StateTr	91.94	89.07	92.32	89.69
StateTr+G2GTr	92.53	90.16	93.07	91.08
BERT StateTr	94.66	91.94	95.18	92.73
BERT StateCLSTr	93.62	90.95	94.31	91.85
BERT StateTr+G2GTr	94.96	92.88	95.58	93.74
BERT StateTr+G2CLSTr	94.29	92.13	94.83	92.96
BERT StateTr+G2GTr+C	94.41	92.25	94.89	92.93
BERT SentTr	95.34	93.29	95.65	93.85
BERT SentTr+G2GTr	**95.66**	**93.60**	**96.06**	**94.26**
BERT SentTr+G2GTr-7 layer	**95.78**	**93.74**	**96.11**	**94.33**

EPFL idiap RESEARCH INSTITUTE

UD Results

- Selected languages contain different:
  - Training size
  - Non-projectivity
  - Morphological feature
  - ….
- Baseline (Kulmizev, 2019) is also using BERT embeddings as an additional input
- Reach state-of-the-art results

Language	Baseline	BERT SentrTr+G2GTr	Relative Error Reduction
Arabic	81.9	83.65	+9.66%
Basque	77.9	83.88	+27.06%
Chinese	83.7	87.49	+23.25%
English	87.8	90.35	+20.90%
Finnish	85.1	89.47	+29.33%
Hebrew	85.5	88.75	+22.41%
Hindi	89.5	93.12	+34.48%
Italian	92	93.99	+24.88%
Japanese	92.9	95.51	+36.76%
Korean	83.7	87.09	+20.80%
Russian	91.5	93.3	+21.18%
Swedish	87.6	90.4	+22.58%
Turkish	64.2	67.77	+9.97%

Conclusions and Future Works

- We proposed a general attention-based architecture (Graph-to-Graph Transformer) to encode both sequences and graphs, and produce output graphs.
- We successfully integrated it with BERT pre-training.
- We achieved state-of-the-art results on transition-based dependency parsing.

- You can easily apply our G2G Transformer to many NLP tasks.
- Also, check out our follow-up paper “**Recursive Non-Autoregressive Graph-to-Graph Transformer for Dependency Parsing with Iterative Refinement**”, accepted to TACL.

Thanks for your consideration

Code:
https://github.com/alirezamshi/G2GTr

More details in the paper

References

- Ryan McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers.Computational Linguistics, 37(1):197–230
- Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. InProceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together, pages 50–57, Barcelona, Spain.Association for Computational Linguistics
- Joakim Nivre. 2009.Non-projective dependency parsing in expected linear time. In 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,pages 351–359, Suntec, Singapore. Association for Computational Linguistics
- Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. 2019. Deep contextualized word embeddings in transition-based and graph-based dependency parsing – a tale of two parsers revisited.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transitionbased dependency parsing with stack long short-term memory. 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 334–343, Beijing, China. Association for Computational Linguistics.
- David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. 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 323–333, Beijing, China. Association for Computational Linguistics.
- James Cross and Liang Huang. 2016. Incremental parsing with minimal features using bi-directional LSTM. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 32–37, Berlin, Germany. Association for Computational Linguistics.
- Miguel Ballesteros, Yoav Goldberg, Chris Dyer, and Noah A Smith. 2016. Training with exploration improves a greedy stack-lstm parser. arXiv preprint arXiv:1603.03793
- Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2442–2452, Berlin, Germany. Association for Computational Linguistics.
- Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. Transactions of the Association for Computational Linguistics, 4:313–327
- Liner Yang, Meishan Zhang, Yang Liu, Nan Yu, Maosong Sun, and Guohong Fu. 2017. Joint pos tagging and dependency parsing with transition-based neural networks.
- Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2017. Stack-based multi-layer attention for transition-based dependency parsing. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1677–1682, Copenhagen, Denmark. Association for Computational Linguistics.
- Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. Seq2seq dependency parsing. In Proceedings of the 27th International Conference on Computational Linguistics, pages 3203–3214, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Alireza Mohammadshahi, James Henderson, 2020, Recursive Non-Autoregressive Graph-to-Graph Transformer for Dependency Parsing with Iterative Refinement, Transactions of the Association for Computational Linguistics.