Enhancing compositional properties of neural networks by incorporating linguistic insights into their structures

Antoine Simoulin

January, 27th 2023

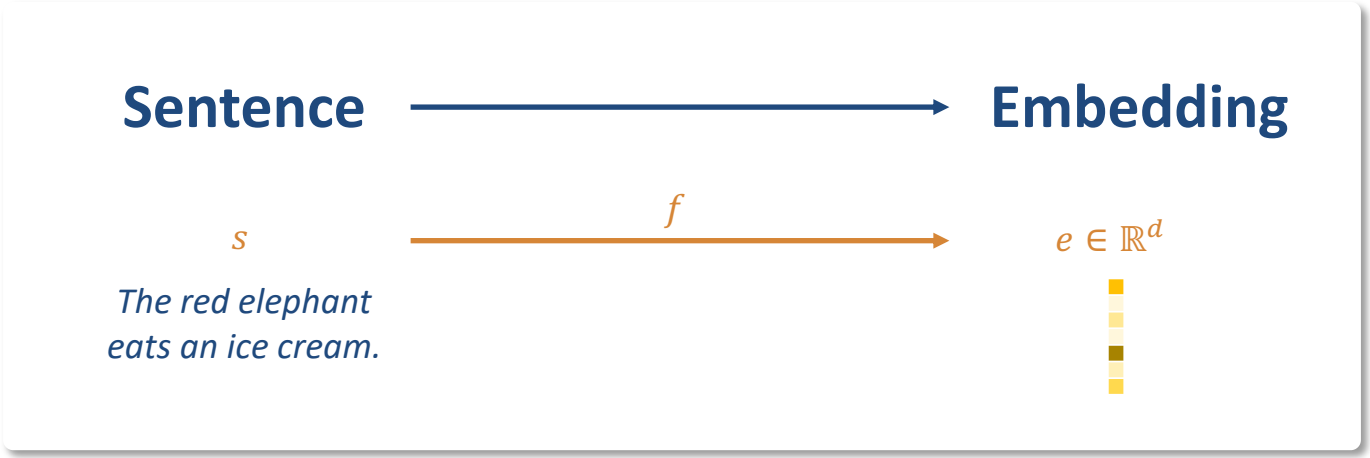
Research Colloquium University of Düsseldorf

Introduction

Introduction

Represent the meaning of sentences

An **embedding** is a **numerical representation** of a word or a sentence in natural language that **preserves the meaning of the statement**. Sentence embeddings have **many applications**: search engines, automatic question answering, text or image generation, etc. [1, 2]



[1] Nils Reimers, Iryna Gurevych: **Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks**. EMNLP/IJCNLP (1) 2019: 3980-3990
[2] Ruiqi Li, Xiang Zhao, and Marie-Francine Moens: **A Brief Overview of Universal Sentence Representation Methods: A Linguistic View.** ACM Comput. Surv. 2022: 55, 3, Article 56

Introduction

Properties of sentence embeddings

The representation space is typically a low-dimensional vector space d. We can translate the **semantic properties** of the initial space by **algebraic transformations** in the representation space.

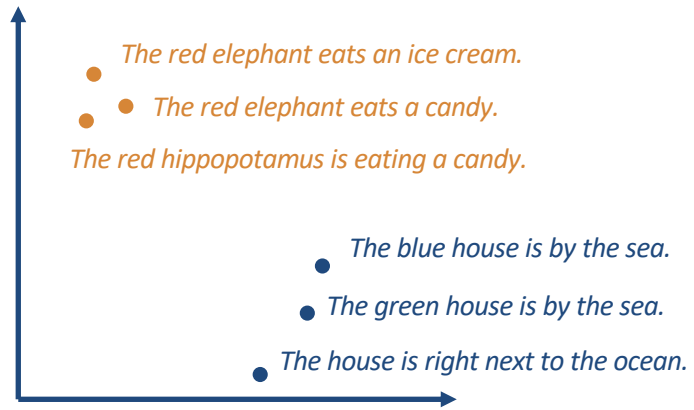


Figure 1. The geometry of the representation space translates the semantic properties between pairs of sentences.

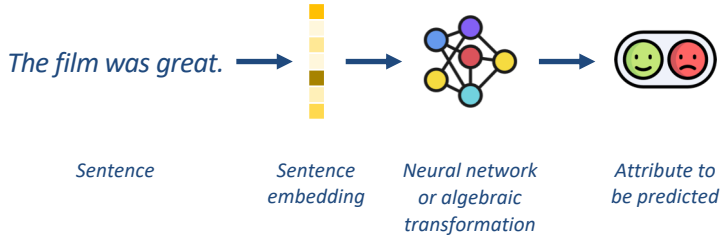
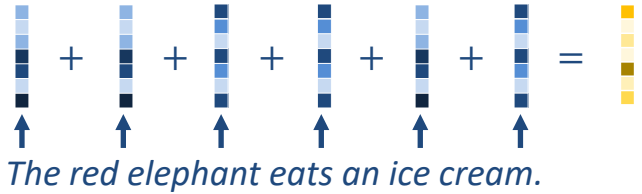


Figure 2. Properties of individual sentences can be inferred from their embeddings using algebraic transformations.

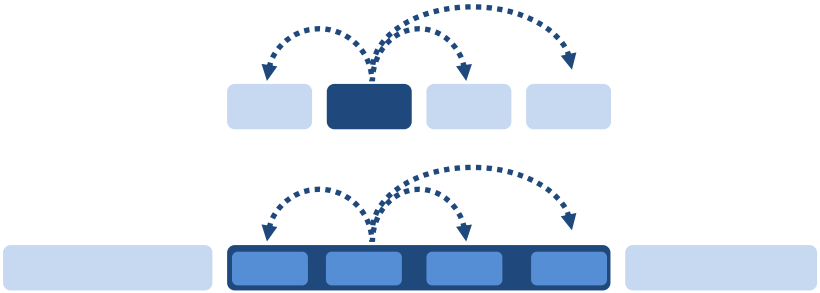
Introduction

From words to sentences

Word embeddings has received wide attention from the scientific community. [1, 2, 3, 4, 5] Nevertheless, **results at the word level are not directly transferable to the sentence level.**



Using BERT's [CLS] or BERT's average final layer embeddings **does not perform well** in the sentence embeddings assessment tasks.



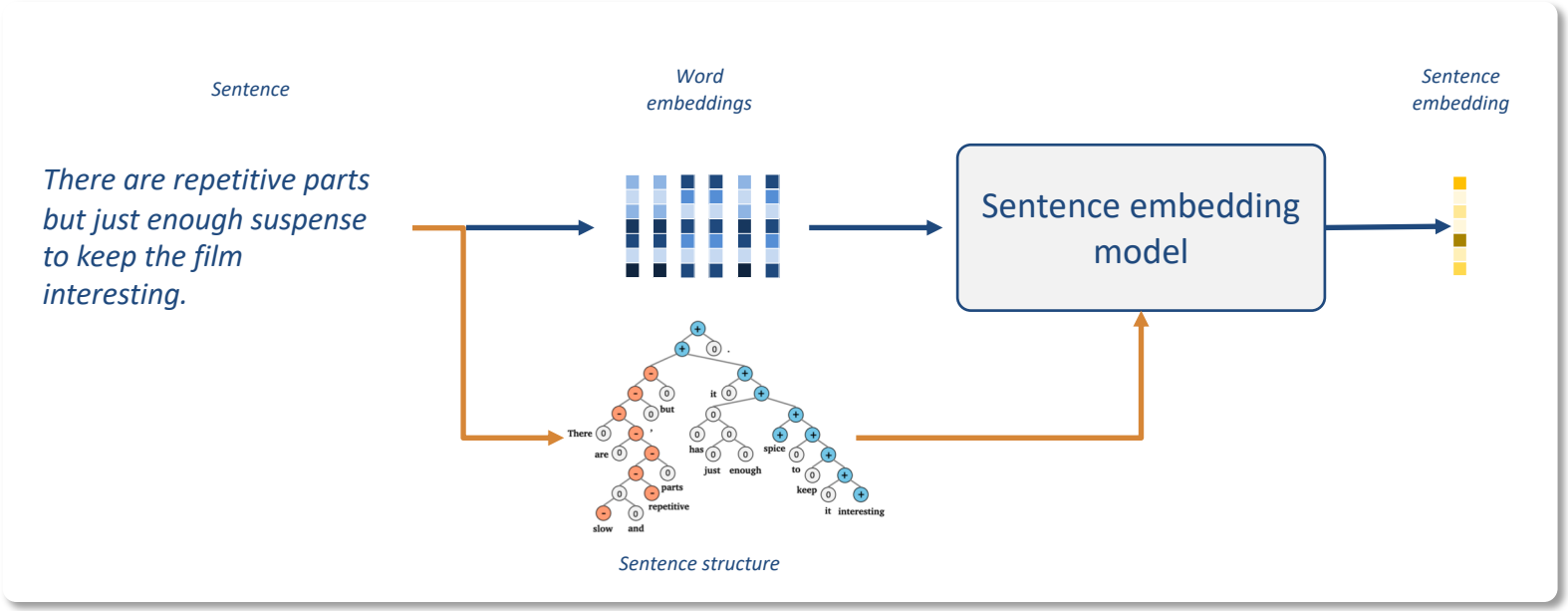
Word embeddings generally assume a fixed vocabulary size and rely on the context of words in a corpus. At the sentence level, **we must take into account the words that make up the sentence and their structure.**

[1] Tomás Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: ***Efficient Estimation of Word Representations in Vector Space.*** ICLR (Workshop Poster) 2013
[2] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, Jeffrey Dean: ***Distributed Representations of Words and Phrases and their Compositionality.*** NIPS 2013: 3111-3119
[3] Jeffrey Pennington, Richard Socher, Christopher D. Manning: ***Glove: Global Vectors for Word Representation.*** EMNLP 2014: 1532-1543
[4] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer: ***Deep Contextualized Word Representations.*** NAACL-HLT 2018: 2227-2237
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova: ***BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*** NAACL-HLT (1) 2019: 4171-4186
[6] Nils Reimers, Iryna Gurevych: ***Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks***. EMNLP/IJCNLP (1) 2019: 3980-3990
[7] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, Lei Li: ***On the Sentence Embeddings from Pre-trained Language Models.*** EMNLP (1) 2020: 9119-9130

Introduction

Syntax

Word embeddings encode only local information, while sentence embeddings seek to encode global information. To compose the meaning of words into a more global meaning representation, it is necessary to **integrate semantic/syntactic information.**



Introduction

Problem statement

The objective of this work is to introduce a linguistic prior within the architecture of neural networks and to measure the relevance of this approach to build sentence embeddings.

Presentation outline

Introduction

1. **Learning sentence embeddings by combining different structures**
2. **Joint learning of structure and composition operations**
3. **Latent structures in Transformers models**
4. **Relationship between compositional properties and neural structures**

Conclusion and future work

1 Learning sentence embeddings by combining different structures

1. Sentence embeddings by combining different structures

Problem statement and working hypotheses

Most methods for constructing sentence embeddings make very simple structural assumptions.

Encoders associated with different syntactic representations **should encode similar semantic information differently**.

These distinct representations should thus be **complementary** and allow for the construction of better embeddings. [1]

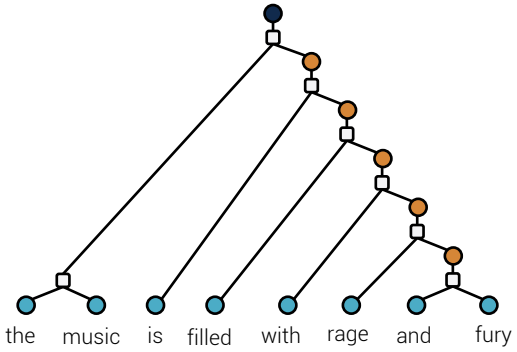
[1] *Antoine Simoulin, Benoît Crabbé:* ***Contrasting distinct structured views to learn sentence embeddings.*** *EACL (student) 2021: 71-79*

1. Sentence embeddings by combining different structures

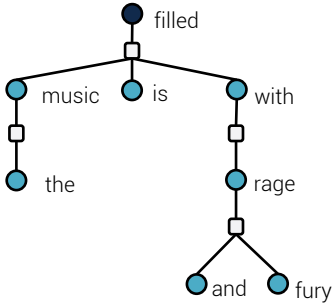
Multi-view learning

We propose a method **combining various syntactic representations** to build sentence embeddings.

SEQ : Sequence and LSTM



CONST : Constituency parsing and N-Ary Tree LSTM



DEP : Dependency parsing and ChildSum Tree LSTM

1. Sentence embeddings by combining different structures

« Contrastive » learning

The model is trained using a **contrastive method.** [1] Given a sentence s, a context sentence s^+, and a set of K negative samples $s_1^- \cdots s_K^-$. The training objective is to **maximize the probability to identify the context sentence among the negative sentences**. With $S = \{s, s^+, s_1^- \cdots s_K^-\}$, f and g two distinct encoders.

$$p(s^+|S) = \frac{e^{c\left(f(s),g(s^+)\right)}}{e^{c\left(f(s),g(s^+)\right)} + \sum_{i=1}^{N} e^{c\left(f(s),g(s_i^-)\right)}} \qquad (1)$$

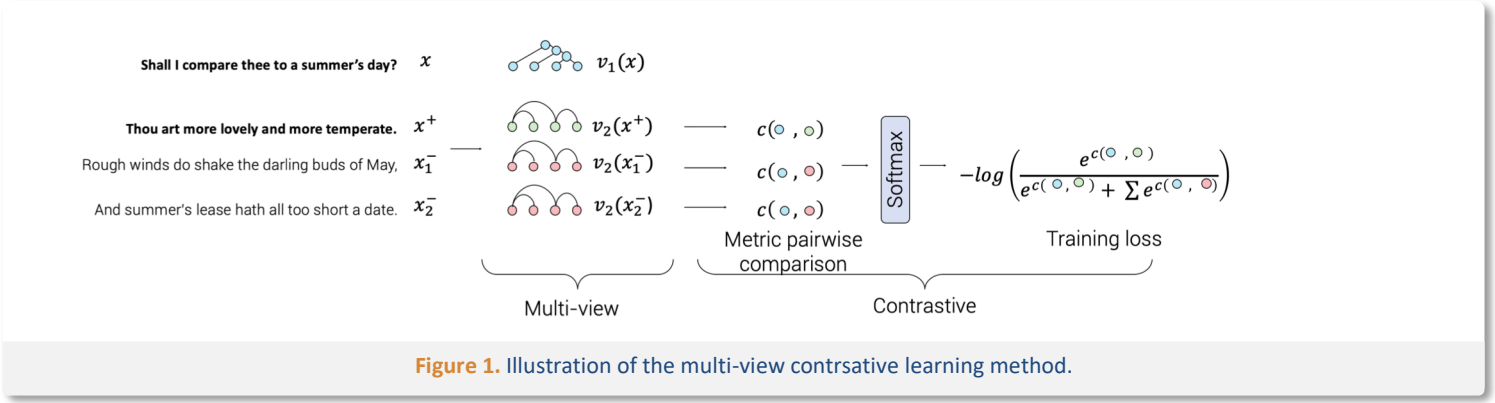


Figure 1. Illustration of the multi-view contrsative learning method.

[1] Lajanugen Logeswaran and Honglak Lee. ***An efficient framework for learning sentence representations.*** *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. 2018*

1. Sentence embeddings by combining different structures

Evaluation on the SentEval benchmark [1]

We report the average score of **SentEval** tasks. Our models expressing a **combination of views give better results** than using a single view.

Model	Avg. SentEval Score
Single-view models	
CONST, CONST	84.4
DEP, DEP	84.6
SEQ, SEQ	84.9
Multi-view models	
SEQ, CONST	85.1
SEQ, DEP	85.3
DEP, CONST	**86.0**

Table 1. Evaluation on the SentEval benchmark

[1] Alexis Conneau, Douwe Kiela: ***SentEval: An Evaluation Toolkit for Universal Sentence Representations.*** LREC 2018

1. Sentence embeddings by combining different structures

Conclusion

In our experience, **multi-view models perform better** than those using a single view.

2 Joint learning of structure and composition operations

2. Joint learning of structure and composition operations

Problem statement and working hypotheses

Recursive models require manually annotated data. However, these resources are not available in all languages.

The optimal structure of an encoder may differ depending on the task.

The optimal **structure of a neural network may differ significantly from that given by linguistic expertise**. The two methods of constructing the semantic representation are very different from a computational point of view. [1]

[1] *Antoine Simoulin and Benoit Crabbé:* ***Unifying Parsing and Tree-Structured Models for Generating Sentence Semantic Representations.*** *NAACL (student) 2022*

Laboratoire de linguistique formelle

Université Paris Cité

Quantmetry

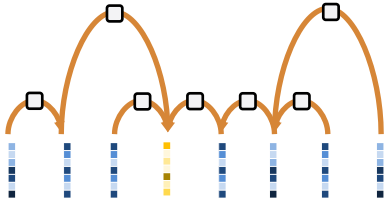
2. Joint learning of structure and composition operations

Latent tree learning

We formulate a **latent tree learning method** [1, 2, 3, 4, 5, 6] that induces trees from raw text and computes semantic representations according to the inferred structure. The sentence folding is predicted by a weighted TREE-LSTM [7] whose tree structure is provided by a dependency parser. [8]

1 A parser **produces a tree structure** from a sentence.

2 A **recursive neural network encodes the sentence** according to the tree structure.

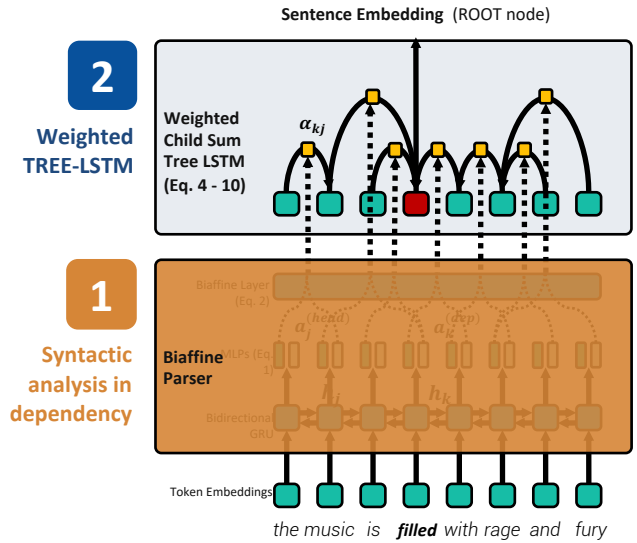


[1] Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, Christopher D. Manning: ***Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions.*** EMNLP 2011: 151-161
[2] Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, Christopher Potts: ***A Fast Unified Model for Parsing and Sentence Understanding.*** ACL (1) 2016
[3] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, Noah A. Smith: ***Recurrent Neural Network Grammars.*** HLT-NAACL 2016: 199-209
[4] Jean Maillard, Stephen Clark, Dani Yogatama: ***Jointly learning sentence embeddings and syntax with unsupervised Tree-LSTMs.*** Nat. Lang. Eng. 25(4): 433-449 (2019)
[5] Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, Wang Ling: ***Learning to Compose Words into Sentences with Reinforcement Learning.*** ICLR (Poster) 2017
[6] Yoon Kim, Alexander M. Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, Gábor Melis: ***Unsupervised Recurrent Neural Network Grammars.*** NAACL-HLT (1) 2019: 1105-1117
[7] Timothy Dozat, Christopher D. Manning: ***Deep Biaffine Attention for Neural Dependency Parsing.*** ICLR (Poster) 2017
[8] Kai Sheng Tai, Richard Socher, Christopher D. Manning: ***Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks.*** ACL (1) 2015: 1556-156

2. Joint learning of structure and composition operations

Method

Our model performs both the syntactic analysis of the sentence and the prediction of its embedding. **The sentence embedding is predicted by a weighted TREE-LSTM [2] whose tree structure is provided by a dependency parser. [1]**



$$\tilde{h}_j = \sum_{k \in C(j)} \alpha_{kj} h_k, \quad (4)$$

$$i_j = \sigma\left(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)}\right), \quad (5)$$

$$o_j = \sigma\left(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)}\right), \quad (6)$$

$$u_j = \tanh\left(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)}\right), \quad (7)$$

$$f_{jk} = \sigma\left(W^{(f)} x_j + U^{(f)} h_k + b^{(f)}\right), \quad (8)$$

$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k, \quad (9)$$

$$h_j = o_j \odot \tanh(c_j), \quad (10)$$

$$\alpha_{kj} = \mathbb{1}_{\text{mst}(s^{(arc)}_{kj})} s^{(arc)}_{kj} \quad (3)$$

$$a^{(dep)}_k = \text{MLP}(h_k), \quad a^{(head)}_j = \text{MLP}(h_j) \quad (1)$$

$$s^{(arc)}_{kj} = (a^{(dep)}_k \oplus 1)^\top U^{(b)} a^{(head)}_j + b^{(b)} \quad (2)$$

Figure 1. The recursive composition function of the TREE-LSTM uses a **weighted sum whose weights are provided by the edges of the parser**, thus linking the outputs of the parser to the recursion of the TREE-LSTM.

*[1] Timothy Dozat, Christopher D. Manning: **Deep Biaffine Attention for Neural Dependency Parsing.** ICLR (Poster) 2017*
*[2] Kai Sheng Tai, Richard Socher, Christopher D. Manning: **Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks.** ACL (1) 2015: 1556-156*

2. Joint learning of structure and composition operations

Experiences (1/2) evaluation on downstream tasks

We first evaluate our model **on the SICK-R task.** [2] We first train the parser submodel on human-annotated sentences from the Penn Tree Bank (PTB). [1] We then refine the parser parameters on the task while training the full model.

Encoder	r
BOW†	78.2 (1,1)
LSTM†	84.6 (0.4)
Bidirectional LSTM†	85.1 (0.4)
N-ary TREE-LSTM† (Tai et al., 2015)	85.3 (0.7)
Childsum TREE-LSTM† (Tai et al., 2015)	86.5 (0.4)
BERT-base (Devlin et al., 2019)	87.3 (0.9)
Unified TREE-LSTM† (Our model)	87.0 (0.3)

Table 1. Evaluation on the **SICK-R task.** We report the Pearson correlation on the test set.

[1] Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz: ***Building a Large Annotated Corpus of English: The Penn Treebank.*** Comput. Linguistics 19(2): 313-330 (1993)
[2] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli: ***A SICK cure for the evaluation of compositional distributional semantic models.*** LREC 2014: 216-223
[3] Kai Sheng Tai, Richard Socher, Christopher D. Manning: ***Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks.*** ACL (1) 2015: 1556-1566
[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova: ***BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.*** NAACL-HLT (1) 2019: 4171-4186

2. Joint learning of structure and composition operations

Experiments (2/2) impact of the initialization of the parser

We study the impact of the **parser initialization on the final performance and structures**. We **initialize the parser using different proportions of the PTB**: the entire PTB (PTB-All), 100 randomly selected samples (PTB-100), or the random initialization of the parser (PTB-Ø).

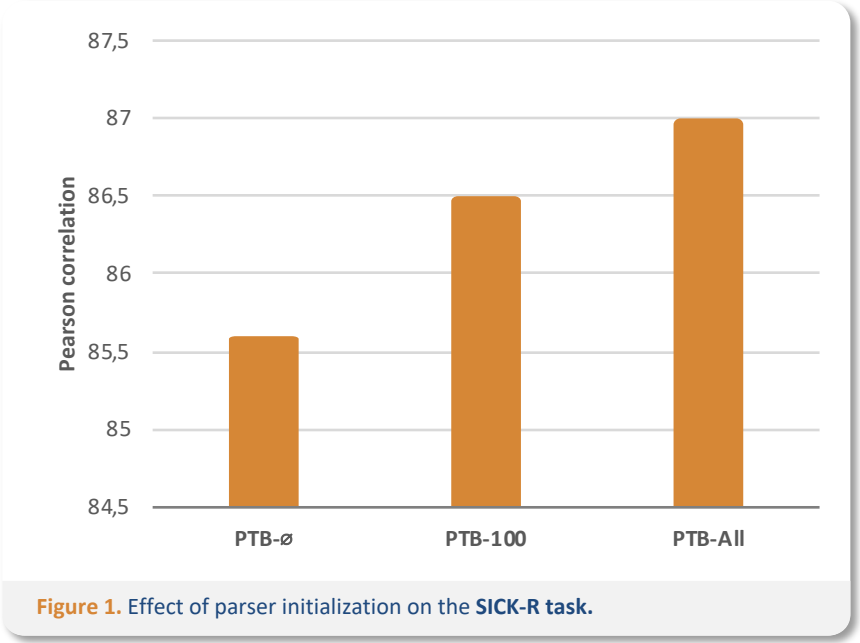


Figure 1. Effect of parser initialization on the **SICK-R task.**

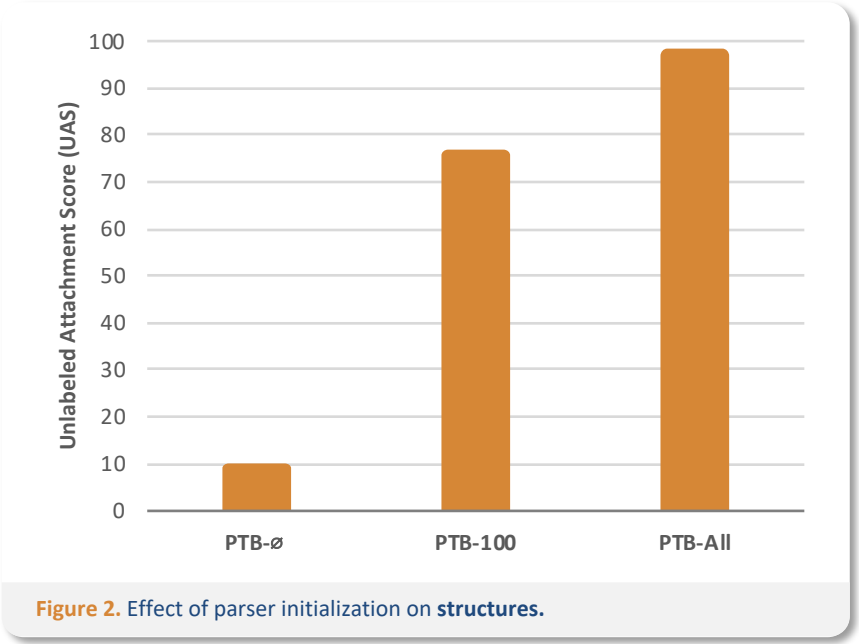


Figure 2. Effect of parser initialization on **structures.**

2. Joint learning of structure and composition operations

Conclusion

We evaluate our model on textual entailment and semantic similarity tasks and **outperform sequential and tree-based models relying on external parsers.**

When initialized on manually annotated structures, our model **improves inference close to BERT's performance on the semantic similarity task.**

We corroborate that the **use of semantic supervision alone is insufficient to produce easily interpretable structures**.

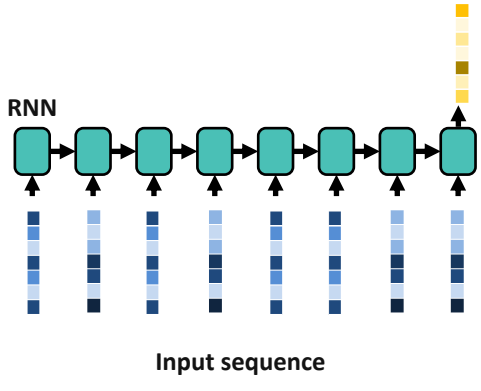
We present additional analyses and experiments in the paper. We propose a proof of concept in which we **use BERT's internal representations to infer sentence structure.**

3 Latent structures in Transformers models

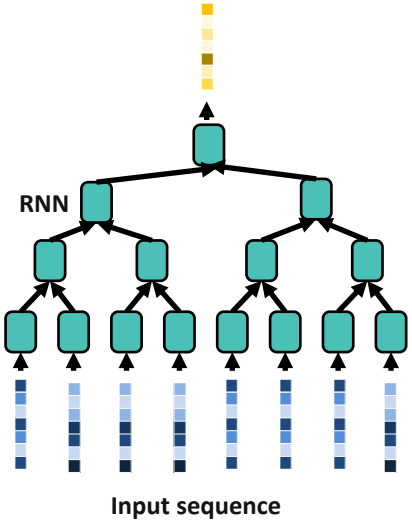
3. Latent structures in Transformers models

Transformer architecture

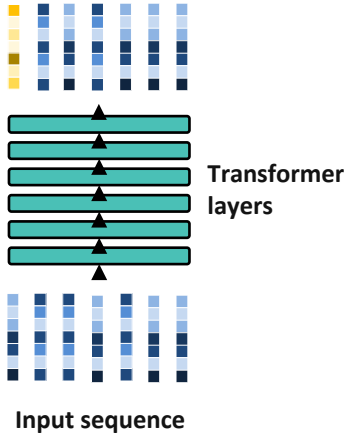
We have used several types of neural network architectures: **sequential, tree-based and graph-based methods.**



Sequential neural network



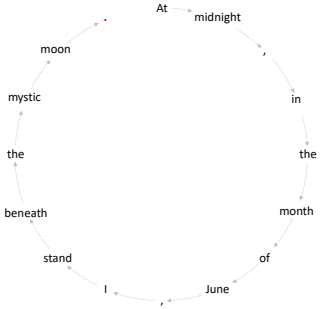
Recursive neural network



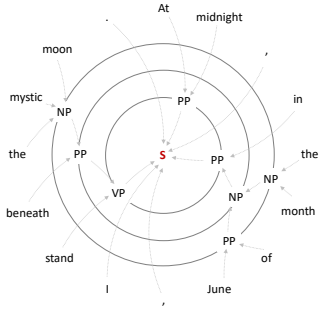
Graph neural network

3. Latent structures in Transformers models

Graphs, trees, sequences

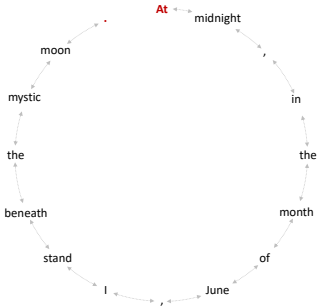


Unidirectional RNN

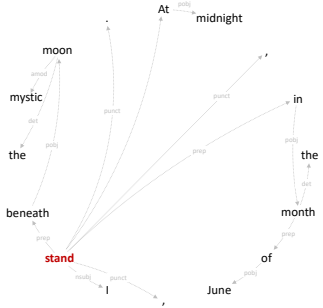


N-ary Tree RNN

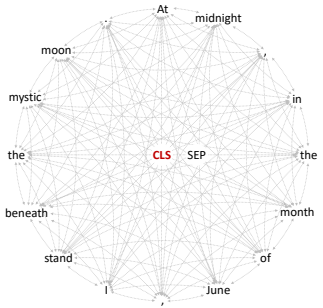
Convolutional NN



Bidirectional RNN



Childsum Tree RNN



Transformer NN

Structure associated with neural networks

3. Latent structures in Transformers models

Problem statement and working hypotheses

The structures learned in the previous chapter remain inspired by linguistic expertise: they follow a dependency parsing formalism. Recursive neural networks can be complex to implement and to optimize from an implementation point of view. Less constrained latent structures could have advantages in terms of ease of implementation. [7]

Transformers are **suspected to be over-parameterized**. [1, 2, 3, 4, 5] Yet the role of this over-parameterization is not well understood. We study the role of multiple layers in deep transformer models.

We interpret the layers as an **iterative process** where contextualized representations of tokens are gradually refined.

We design a variant of ALBERT [6] that **dynamically adapts the number of layers for each token in the input.**

[1] Daoyuan Chen, Yaliang Li, Minghui Qiu, Zhen Wang, Bofang Li, Bolin Ding, Hongbo Deng, Jun Huang, Wei Lin, Jingren Zhou: ***AdaBERT: Task-Adaptive BERT Compression with Differentiable Neural Architecture Search.*** IJCAI 2020: 2463-2469

[2] Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, Qun Liu: ***DynaBERT: Dynamic BERT with Adaptive Width and Depth.*** NeurIPS 2020

[3] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, Ivan Titov: ***Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned.*** ACL (1) 2019: 5797-5808

[4] Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, Qi Ju: ***FastBERT: a Self-distilling BERT with Adaptive Inference Time.*** ACL 2020: 6035-6044

[5] Angela Fan, Edouard Grave, Armand Joulin: ***Reducing Transformer Depth on Demand with Structured Dropout.*** ICLR 2020

[6] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut: ***ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.*** ICLR 2020

[7] Antoine Simoulin, Benoît Crabbé: How Many Layers and Why? An Analysis of the Model Depth in Transformers. ACL (student) 2021: 221-228

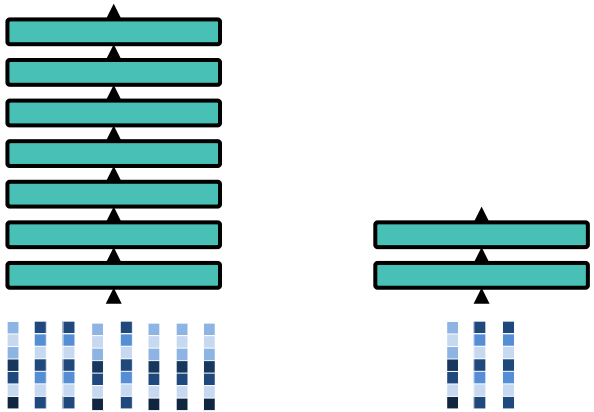
3. Latent structures in Transformers models

Method

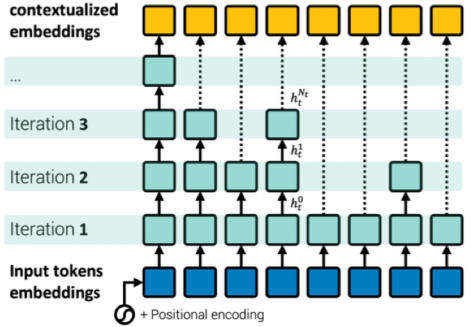
Some models propose to **dynamically adapt the number of layers to the complexity of the sentence**. *[1, 2, 3]*

Many layers may be required to extract semantic information from a complete complex paragraph...

... but may be disproportionate to deal with a 5 word sentence.



The model can **adapt the number of layers** applied for each token using an adaptive computing mechanism. *[4]*



[1] Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, Qi Ju: ***FastBERT: a Self-distilling BERT with Adaptive Inference Time.*** ACL 2020: 6035-6044

[2] Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian J. McAuley, Ke Xu, Furu Wei: ***BERT Loses Patience: Fast and Robust Inference with Early Exit.*** NeurIPS 2020

[3] Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, Jimmy Lin: ***DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference.*** ACL 2020: 2246-2251

[4] Alex Graves: ***Adaptive Computation Time for Recurrent Neural Networks.*** CoRR abs/1603.08983 (2016)

3. Latent structures in Transformers models

Experiences (1/2) Analysis of the pre-training

Iteration analysis: the model distributes iterations for crucial or difficult tokens. [CLS] and [MASK] receive more iterations than other tokens while [SEP] receives less.

Natural fixed point: By design, we encourage the model to converge to a fixed point. **All tokens reach a fixed point by following a distinct convergence pattern.**

Models	*tiny*	*small*	*base*
τ	1e-3	5e-4	2.5e-4
Max iterations	6	12	24
mlm (Acc.)	55.4	57.1	57.4
sop (Acc.)	80.9	83.9	84.3
All tokens	3.8	7.1	10.0
All unmasked tokens	3.5	6.5	9.2
[MASK/MASK]	5.8	10.9	16.0
[MASK/random]	5.8	10.9	16.0
[MASK/original]	4.0	7.4	10.5
[CLS]	6.0	12.0	22.5
[SEP]	2.5	7.6	8.4

Table 1. Average number of iterations as a function of token types during pre-training. For each model, we report an average number of iterations on our development set at the end of pre-training.

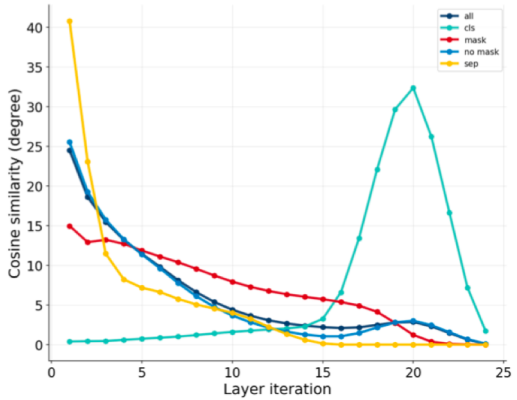


Figure 1. Evolution of cosine similarity between hidden states h^n and h^{n+1} from two consecutive iterations. We measure the iterations on our development set at the end of the pre-training.

3. Latent structures in Transformers models

Experiences (2/2) downstream tasks

Probing tasks: We evaluate the model on linguistic probes [2] and **compare the number of iterations performed for each token according to its dependency type.**

Control test : We evaluate our configuration on the GLUE benchmark. [1]

	Tense	Subj Num	Obj Num	Top Const
punct (121k)	5.0	4.8	5.2	6.7
prep (101k)	4.6	4.6	5.4	6.2
pobj (98k)	4.5	4.6	5.4	5.8
det (86k)	4.5	4.6	5.1	6.1
nn (81k)	5.1	5.4	**5.8**	6.7
nsubj (80k)	**5.3**	**6.1**	**5.9**	**7.5**
amod (66k)	4.6	4.9	5.5	6.1
dobj (49k)	4.8	5.0	**5.9**	6.1
root (44k)	**5.9**	**6.1**	**6.2**	**7.9**
advmod (37k)	4.8	4.8	5.3	6.8
avg.	5.4	5.4	5.8	7.2
test Acc.	87.5	93.9	96.1	91.2
baseline Acc.	87.3	94.0	96.0	91.9

Table 1. Distribution of iterations according to token dependency types.

	Avg. Glue score
BERT-base	76.9
ALBERT-base	75.6
ALBERT-base + Adapt. Depth	75.2
ALBERT-small + Adapt. Depth	74.2
ALBERT-tiny + Adapt. Depth	72.6

Table 2. GLUE test results.

*[1] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman: **GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding.** ICLR (Poster) 2019*
*[2] Alexis Conneau, Douwe Kiela: **SentEval: An Evaluation Toolkit for Universal Sentence Representations.** LREC 2018*

3. Latent structures in Transformers models

Conclusion

Deep transform models are over-parameterized, and most of their parameters are redundant.

A better understanding of the internal mechanism in place in deep transformer models appears to be a necessity to reduce computational power or **provide new regularization methods.**

Our experiments provide a new avenue of interpretation for the role of layers in deep transformer models: **layers could be interpreted as the iteration of an iterative process.**

4 Relationship between compositional properties and neural structures

4. Relationship between compositional properties and neural structures

Problem statement and working hypotheses

In the previous chapters, we have proposed different neural network architectures with various degrees of constraints. We have evaluated for each of them their contribution to solve tasks requiring to encode the global semantics of the sentence. We now seek to refine the evaluation to quantify the impact on the compositional properties.

Linguistic models can **rely on less systematic phenomena** such as lexical biases or superficial heuristics in the data to accomplish the task. [1]

It is definitely **possible to train efficient language models without prior compositional properties.**

Evaluating the compositional properties of models is notoriously difficult. **Data creation is a critical step.**

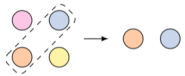
[1] Tal Linzen, Brian Leonard: ***Distinct patterns of syntactic agreement errors in recurrent networks and humans.*** *CogSci 2018*

4. Relationship between compositional properties and neural structures

Method

We introduce **CobA**, **Co**mpositionality **b**enchmark using **A**rithmetic. It is easy to build arithmetic examples by isolating specific properties. We train the model to evaluate expressions in the domain. We then infer expressions from the generalization set which includes partitions for **Systematicity, Productivity, Substitutability, and Localism.**

Systematicity
Recombination of known parts to form new sequences.



Training

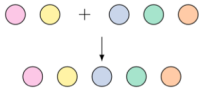
12×4

7×33

Evaluation

12×7

Productivity
Extrapolation to longer sequences.



Training

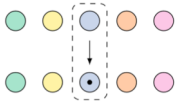
$3 + 4$

$5 + 4$

Evaluation

$4 + 4 + 2 + 5$

Substitutivity
Robustness in the introduction of synonyms.



Training

3×4

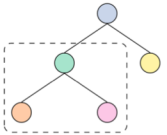
5×4

Evaluation

6×7

7×6

Localism
Recursive evaluation of small constituents before large constituents.



Training

3×4

5×4

Evaluation

6×12

$(2 + 4) \times 12$

*[1] Dieuwke Hupkes, Verna Dankers, Mathijs Mul, Elia Bruni: **Compositionality Decomposed: How do Neural Networks Generalise?** J. Artif. Intell. Res. 67: 757-795 (2020)*

4. Relationship between compositional properties and neural structures

Experiences (1/2) evaluation tasks

In general, models based on LSTM cells perform better than transformers. For productivity and systematicity, sequential models strongly outperform models using a fixed length context. As for transformers, we confirm their known limitation in terms of productivity.

Encoders	# parameters ($\times 1e^3$)	Random	Localism	Systematicity	Productivity	Substitutivity
Random	—	39.5 (0.1)	40.1 (0.1)	40.0 (0.2)	39.1 (0.2)	40.0 (0.2)
BoW	68	28.7 (8.1)	20.6 (4.8)	37.0 (0.2)	38.3 (5.4)	19.0 (0.5)
LSTM (uni)	595	3.5 (0.4)	7.3 (0.8)	2.7 (0.3)	14.1 (0.4)	2.4 (0.3)
LSTM (bi)	1,186	2.2 (0.2)	8.6 (1.0)	2.5 (0.3)	13.5 (1.1)	1.3 (0.1)
LSTM (tree)	1,012	5.4 (0.1)	8.9 (0.1)	7.4 (0.6)	15.0 (0.5)	4.9 (0.3)
Transformer (BERT)	1,290	3.6 (1.1)	11.9 (0.7)	20.6 (4.9)	30.2 (1.8)	2.3 (0.8)
Transformer (ALBERT)	315	6.9 (1.5)	13.0 (0.9)	16.6 (6.6)	25.9 (3.1)	4.8 (1.0)

Table 1. Evaluation of compositionality.

4. Relationship between compositional properties and neural structures

Experiments (2/2) Impact of sentence length

Productivity is notoriously difficult. We decompose the productivity generalization set as a function of the number of operators and the number of exposure instances and plot the evolution of the RMSE.

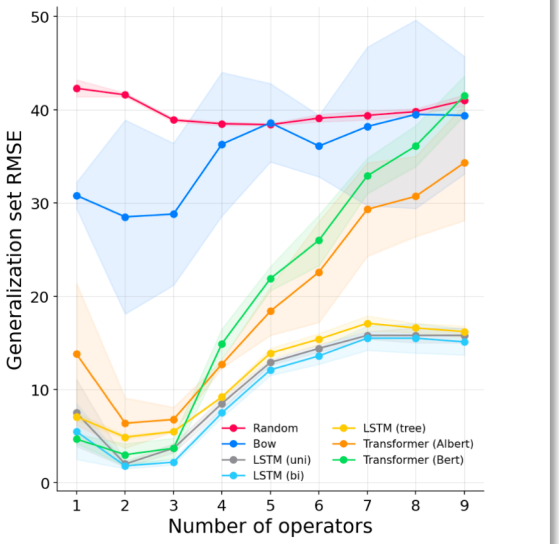


Figure 1. Evolution of RMSE for productivity as a function of the **number of operators.**

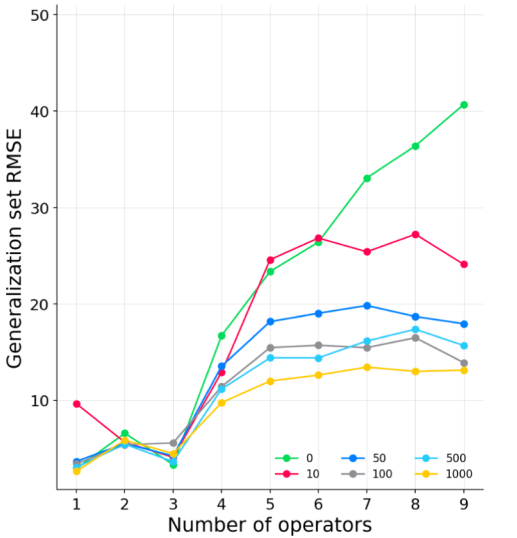


Figure 2. Evolution of the RMSE for productivity as a function of the **number of exposure examples.**

4. Relationship between compositional properties and neural structures

Conclusion

In general, the models are **robust to the introduction of paraphrases** (substitutability) and **can perform recursive evaluation of subcomponents** (localism).

Transformers struggle to generalize to longer sequences (productivity) **or to combine known parts to form new sequences** (systematicity).

The models **struggle with complex expressions, involving more symbols, more complex structure, or more operator types**. We slightly improve the degree of compositionality by adjusting the number of parameters or exposure to generalization expressions.

Conclusion and future work

Conclusion

The introduction of a **strong structure prior** in neural networks allows in some configurations to **build better representations**. In particular by combining various structures or by learning the structure at the same time as the composition function.

On many tasks, **transformers**, which make very few structural assumptions, continue to obtain state-of-the-art results. We propose an **interpretation of these networks according to a graph formalism**.

A finer evaluation of the compositional properties of the models and highlight the limitation of transformers on some key properties.

In a second part we compared the approaches to structure modeling by **scaling the models and the data**.

We propose **several resources**: **corpora** for compositionality evaluation or model generation capabilities in French, **pre-trained models** in French, state-of-the-art sentence embedding models pre-trained in English, recursive network implementation **libraries**.

Contributions

Conference papers

Antoine Simoulin, Benoît Crabbé: ***Contrasting distinct structured views to learn sentence embeddings.*** *EACL (student) 2021*
Antoine Simoulin, Benoît Crabbé: ***How Many Layers and Why? An Analysis of the Model Depth in Transformers.*** *ACL (student) 2021*
Antoine Simoulin, Benoit Crabbé: ***Un modèle Transformer Génératif Pré-entrainé pour le _____ français.*** *JEP/TALN/RECITAL 2021*
Antoine Simoulin, Benoit Crabbé: ***Unifying Parsing and Tree-Structured Models for Generating Sentence Semantic Representations.*** *NAACL (student) 2022*

Software

PyTree: *PyTorch Annual Hackathon 2021 WINNER Honorable Mention*
Sentence-Transformers: *Hugging Face JAX Global Sprint – Special Nominee*

Resources

GPT-fr *: https://huggingface.co/asi/gpt-fr-cased-base*
Wikitext-fr *: https://huggingface.co/datasets/asi/wikitext_fr*

Questions ?