Université Paris Cité Sciences Mathématiques de Paris Centre (ED 386)
Laboratoire de Linguistique Formelle (LLF)
Quantmetry

Plongements de phrases et leurs relations avec les structures de phrases

Sentence embeddings and their relation with sentence structures

Antoine Simoulin

7 Juillet 2022

Soutenance de thèse de doctorat d'informatique
Dirigée par Benoit Crabbé

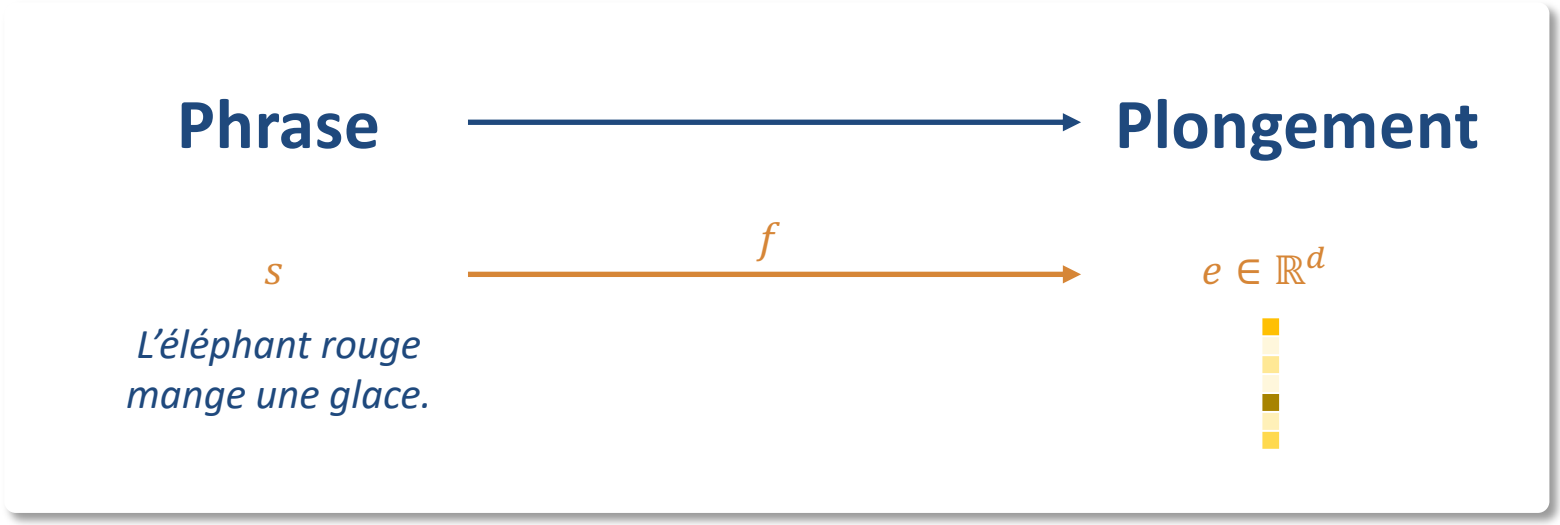
Laboratoire de linguistique formelle
Université Paris Cité
Quantmetry

Introduction

Introduction

Représenter le sens des phrases

Un **plongement** (en anglais ***embedding***) est une **représentation numérique d'un mot, d'une phrase** en langue naturelle qui **conserve le sens de l'énoncé.** Les plongements de phrases ont de **nombreuses applications** : moteur de recherche, réponse automatique aux questions, génération de textes ou d'images, etc [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

Propriétés des plongements de phrases

L'espace de représentation est typiquement un espace vectoriel de faible dimension d. On peut traduire les **propriétés sémantiques** de l'espace initial par des **transformations algébriques** dans l'espace de représentation.

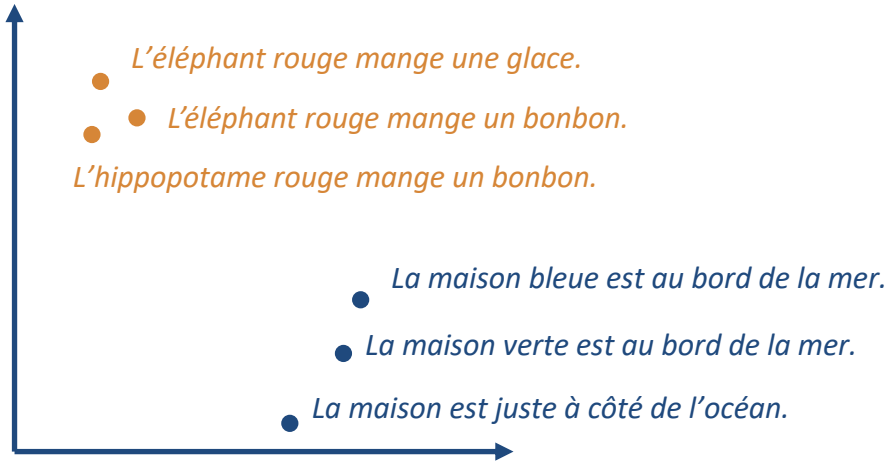


Figure 1. La géométrie de l'espace de représentation traduit les propriétés sémantiques entre des paires de phrases.

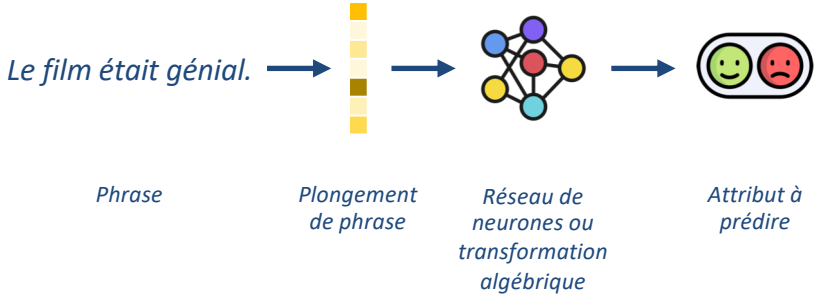
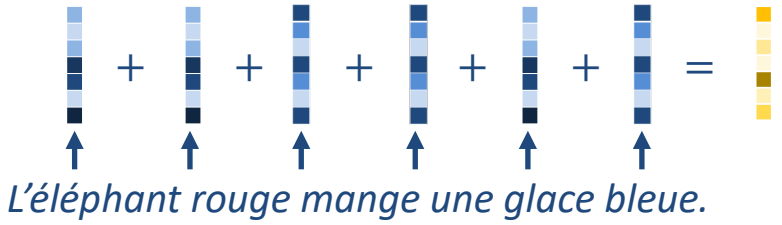


Figure 2. On peut **inférer des propriétés de phrases individuelles** à partir de leur plongement en utilisant des transformations algébriques.

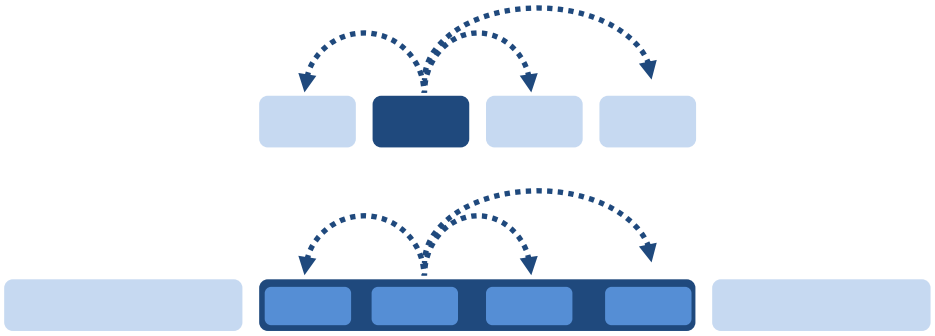
Introduction

Des mots à la phrase

Les plongements de mots bénéficient d'une large attention de la communauté scientifique [1, 2, 3, 4, 5]. Néanmoins, les **résultats à l'échelle du mot ne sont pas directement transposables à l'échelle de la phrase**.



L'utilisation du [CLS] de BERT ou la moyenne des plongements de mots obtenus par BERT n'obtiennent **pas de bonnes performances** dans les tâches d'évaluation des plongements de phrases [6, 7].



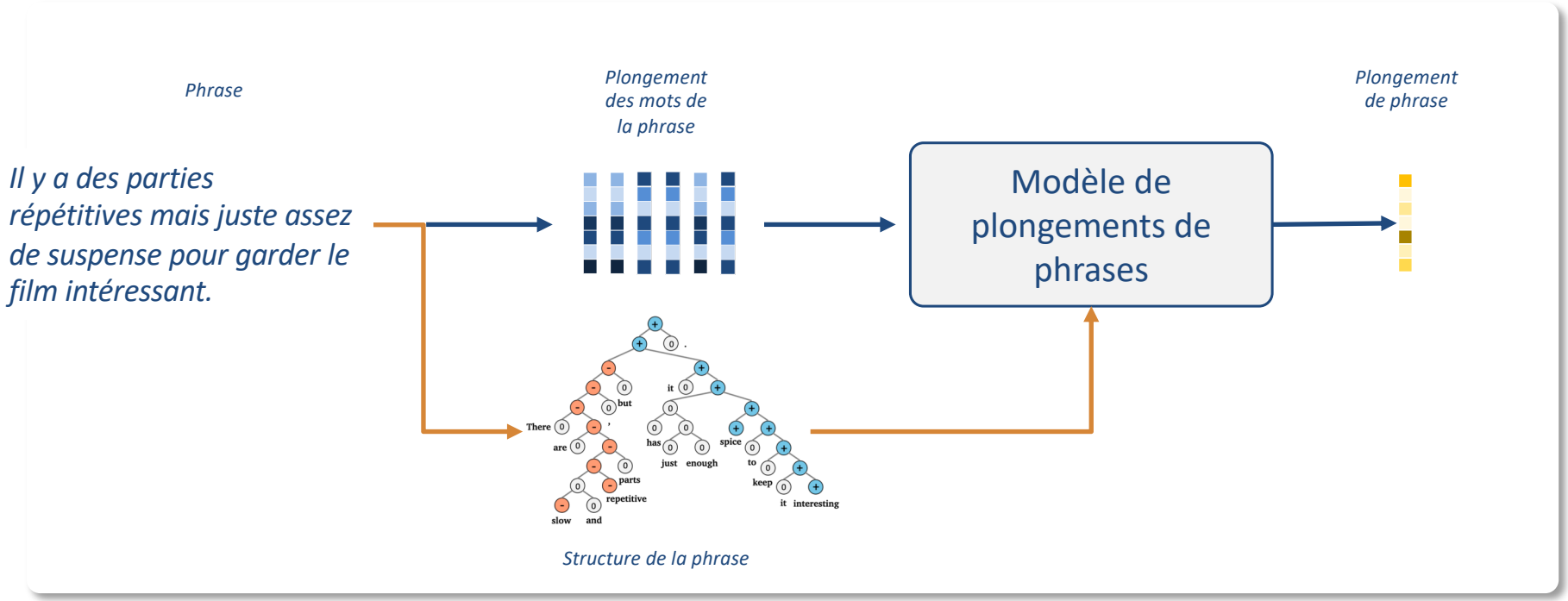
Les plongements de mots supposent généralement un vocabulaire de taille fixe et s'appuient sur le contexte des mots dans un corpus. **A l'échelle des phrases, on doit tenir compte des mots qui la composent et de leur structure.**

[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

La syntaxe

Les plongements de mots ne codent que des informations locales, les plongements de phrases cherchent quant à eux à encoder des informations globales. Pour composer le sens des mots en une représentation de sens plus globale, il est **nécessaire d'intégrer des informations sémantiques/syntaxiques**.



Introduction

Problématique

L'objectif de la thèse est d'introduire un **a priori linguistique** dans **l'architecture des réseaux de neurones** et de mesurer l'intérêt de cette approche pour construire des **plongements de phrases**.

Plan

Introduction

1. **Apprentissage de plongements de phrases en combinant différentes structures**
2. **Apprentissage conjoint de la structure et des opérations de composition**
3. **Structures latentes dans les modèles de transformers**
4. **Relations entre les propriétés compositionnelles et les structures neuronales**

Conclusion et travaux futurs

1 Apprentissage de plongements de phrases en combinant différentes structures

1. Plongements de phrases par combinaison de différentes structures

Problématique et hypothèses de travail

La plupart des méthodes de construction de plongements de phrases font des hypothèses de structures très simples.

Des encodeurs associés à différentes représentations syntaxiques, devraient **encoder différemment des informations sémantiques** similaires.

Ces représentations distinctes devraient ainsi être **complémentaires** et permettre de construire de meilleurs plongements [1].

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

Laboratoire de linguistique formelle

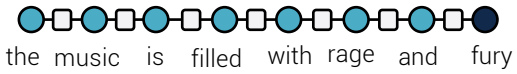
Université Paris Cité

Quantmetry

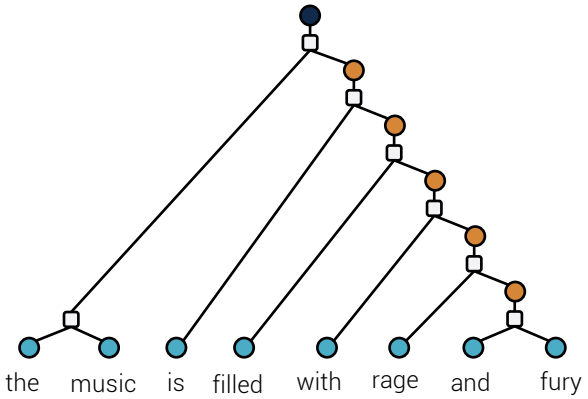
1. Plongements de phrases par combinaison de différentes structures

Apprentissage « multi-vues »

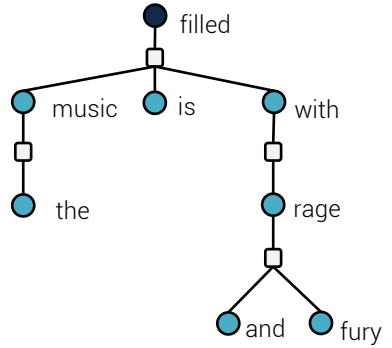
Nous proposons une **méthode combinant diverses représentations syntaxiques** pour construire des plongements de phrases.



SEQ : Séquence et LSTM



CONST : Analyse en constituants et N-Ary Tree LSTM



DEP : Analyse en dépendances et ChildSum Tree LSTM

1. Plongements de phrases par combinaison de différentes structures

Apprentissage « contrastif »

On entraine le modèle en utilisant une **méthode contrastive** [1]. Etant donné une phrase s, une phrase dans le même contexte s^+ et un ensemble de K exemples négatifs $s_1^- \cdots s_K^-$. L'objectif d'entrainement est de **maximiser la probabilité d'identifier la phrase du contexte parmi les exemples négatifs**. Avec $S = \{s, s^+, s_1^- \cdots s_K^-\}$, f et g deux encodeurs distincts.

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

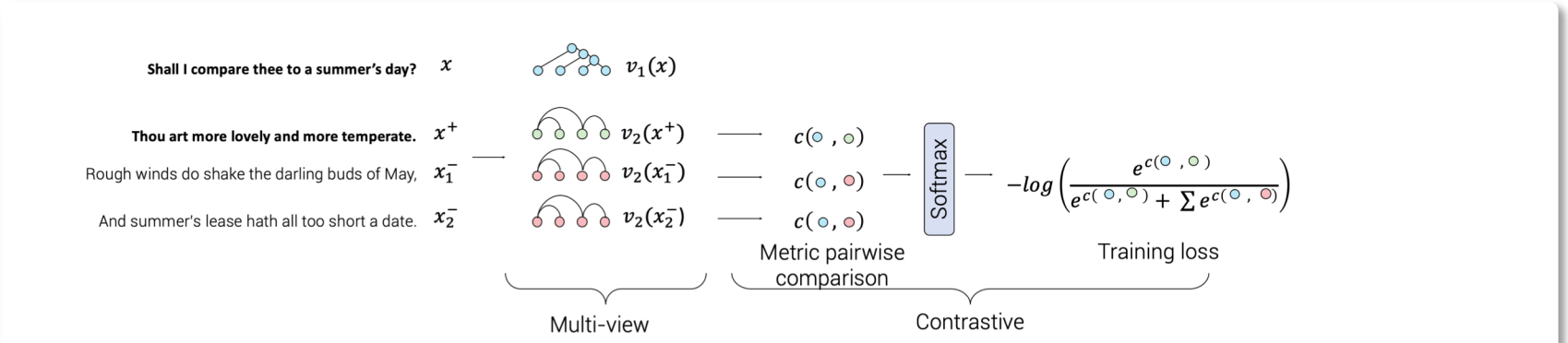


Figure 1. Illustration de l'apprentissage multi-vues contrastif.

[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. Plongements de phrases par combinaison de différentes structures

Evaluation sur le benchmark SentEval [1]

Nous rapportons le score moyen des tâches **SentEval**. Nos modèles exprimant une **combinaison de vues donnent de meilleurs résultats que l'utilisation d'une même vue**.

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 de la méthode sur le benchmark d'évaluation SentEval

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

Figure 2. Structure of the language views.

Laboratoire de linguistique formelle — Université Paris Cité — Quantmetry

1. Plongements de phrases par combinaison de différentes structures

Conclusion

D'après notre expérience, les modèles **multi-vues sont plus performants** que ceux utilisant une seule vue.

2 Apprentissage conjoint de la structure et des opérations de composition

2. Apprentissage conjoint de la structure et des opérations de composition

Problématique et hypothèses de travail

Les modèles récursifs nécessitent des données manuellement annotées. Cependant, ces ressources ne sont pas disponibles dans toutes les langues.

La structure optimale d'un encoder peut différer selon la tâche.

La **structure optimale d'un réseau de neurone peut différer de manière significative de celle donnée par l'expertise linguistique**. Les deux méthodes de construction de la représentation sémantique sont très différentes d'un point de vue calculatoire [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

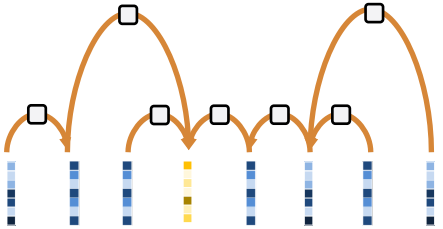
2. Apprentissage conjoint de la structure et des opérations de composition

Apprentissage d'arbres latents

Nous formulons une **méthode d'apprentissage d'arbres latents** [1, 2, 3, 4, 5, 6] qui induit des arbres à partir de texte brut et calcule des représentations sémantiques selon la structure inférée. Le plongement de la phrase est prédit par un TREE-LSTM pondéré [7] dont la structure arborescente est fournie par un analyseur de dépendances [8].

1 Un **analyseur syntaxique produit une structure arborescente** à partir d'une phrase.

2 Un **réseau neuronal récursif encode la phrase** suivant la structure arborescente.



[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. Apprentissage conjoint de la structure et des opérations de composition

Méthode

Notre modèle effectue conjointement l'analyse syntaxique de la phrase et la prédiction de son plongement. **Le plongement de la phrase est prédit par un TREE-LSTM [2] pondéré dont la structure arborescente est fournie par un analyseur de dépendance [1].**

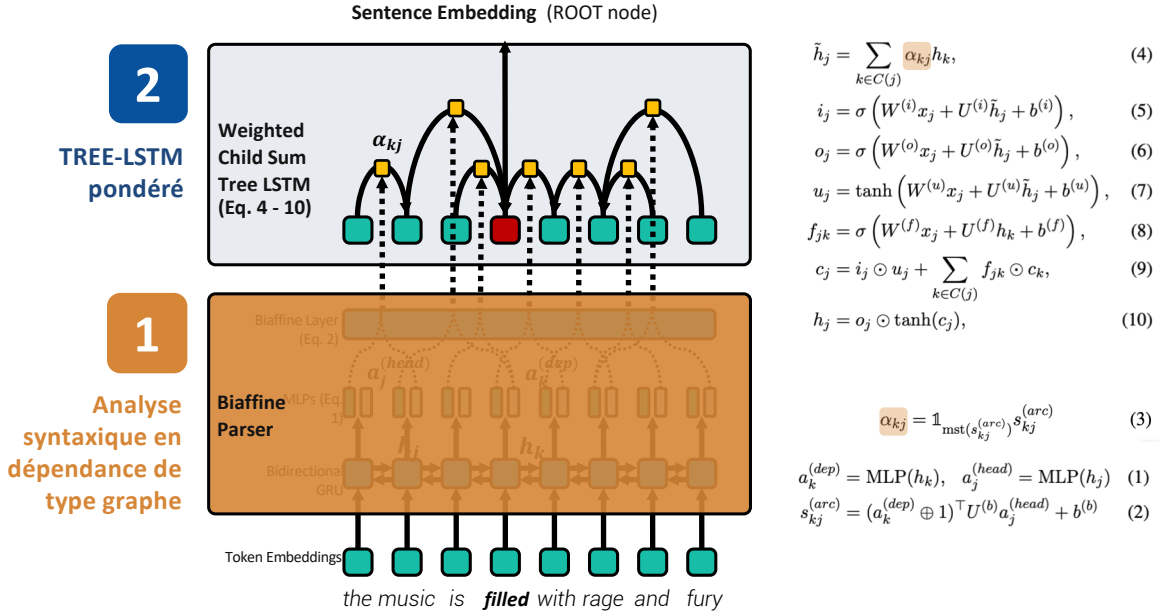


Figure 1. La fonction de composition récursive du TREE-LSTM utilise **une somme pondérée dont les poids sont fournis par les bords de l'analyseur**, reliant ainsi les sorties de l'analyseur à la récursion du TREE-LSTM.

*[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. Apprentissage conjoint de la structure et des opérations de composition

Expériences (1/2) évaluation sur des tâches d'applications

Nous **évaluons d'abord notre modèle sur la tâche SICK-R** [2]. Nous entraînons d'abord le sous-modèle d'analyse syntaxique sur des phrases annotées par des humains provenant du Penn Tree Bank (PTB) [1]. Nous affinons ensuite les paramètres de l'analyseur syntaxique sur la tâche tout en entraînant le modèle complet.

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 sur la **tâche SICK-R.** Nous rapportons la corrélation de Pearson sur l'ensemble de test.

[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. Apprentissage conjoint de la structure et des opérations de composition

Expériences (2/2) impact de l'initialisation de l'analyseur syntaxique

Nous étudions **l'impact de l'initialisation de l'analyseur syntaxique sur la performance finale et sur les structures**. Nous **initialisons l'analyseur syntaxique en utilisant différentes proportions du PTB** : l'ensemble du PTB (PTB-All), 100 échantillons sélectionnés aléatoirement (PTB-100) ou l'initialisation aléatoire de l'analyseur syntaxique (PTB-Ø).

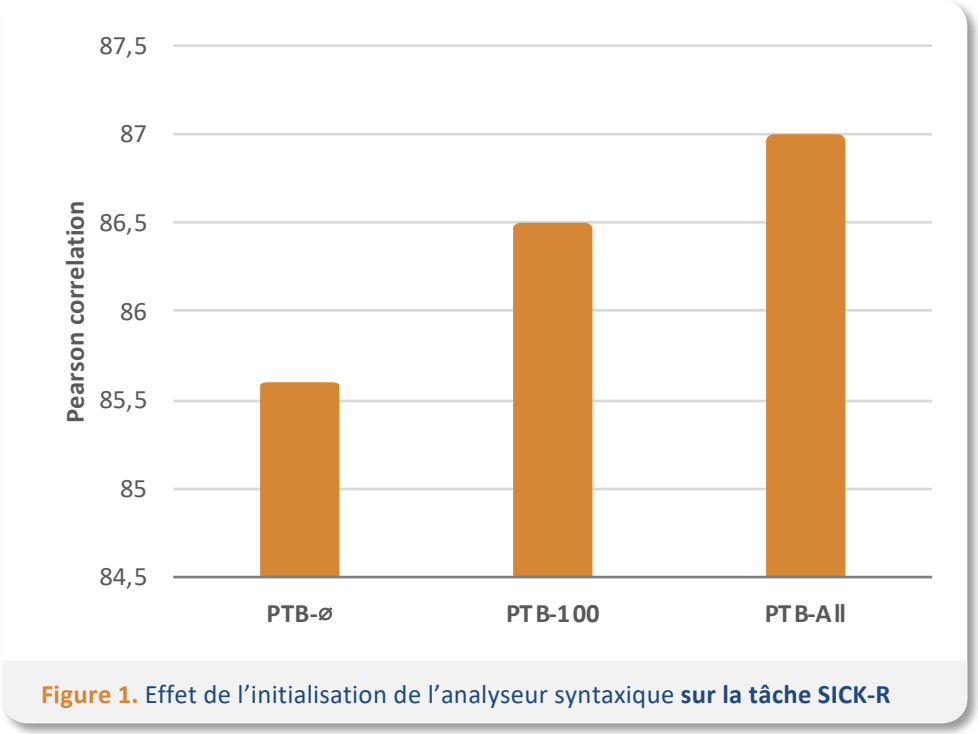


Figure 1. Effet de l'initialisation de l'analyseur syntaxique **sur la tâche SICK-R**

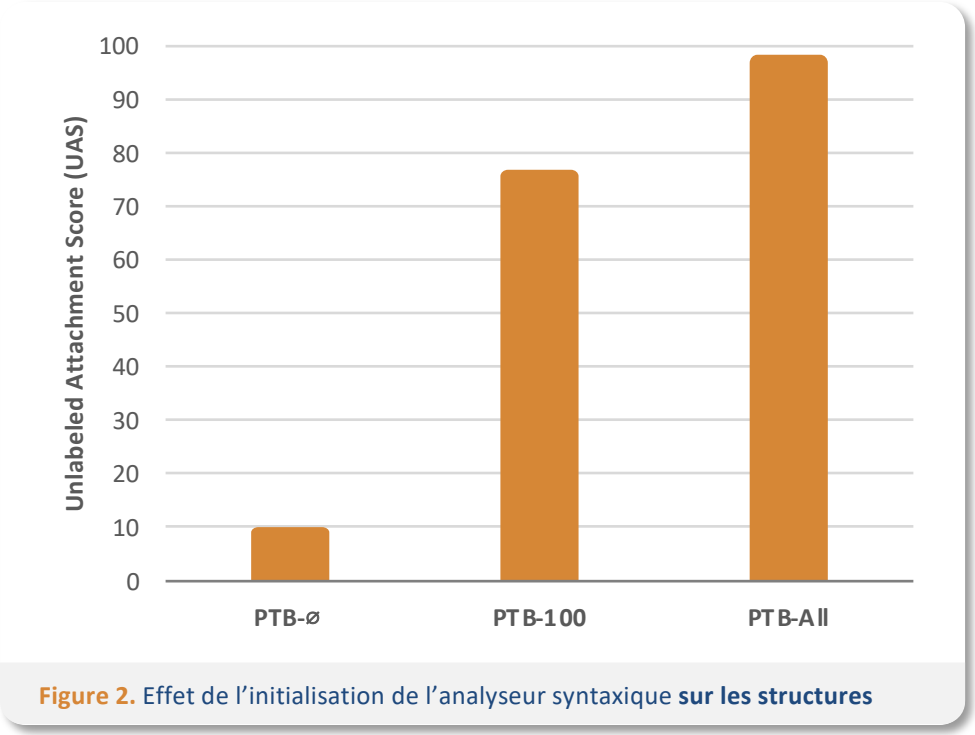


Figure 2. Effet de l'initialisation de l'analyseur syntaxique **sur les structures**

2. Apprentissage conjoint de la structure et des opérations de composition

Conclusion

Nous évaluons notre modèle sur des tâches d'implication textuelle et de similarité sémantique et **nous surpassons les modèles séquentiels et les modèles arborescents reposant sur des analyseurs syntaxiques externes**.

Lorsqu'il est initialisé sur des structures annotées manuellement, notre modèle améliore **l'inférence proche de la performance de BERT** sur la tâche de similarité sémantique.

Nous corroborons que la **seule utilisation de la supervision sémantique est insuffisante pour produire des structures faciles à interpréter.**

Nous présentons des analyses et des expériences supplémentaires dans le manuscrit. Nous proposons une preuve de concept dans laquelle **nous utilisons les représentations internes de BERT pour inférer la structure des phrases**.

3 Structures latentes dans les modèles de transformers

3. Structures latentes dans les modèles de transformers

Problématique et hypothèses de travail

Les structures apprises dans le chapitre précédent restent inspirées par une expertise linguistique : elles suivent un formalisme d'analyse syntaxique en dépendance. Les réseaux de neurones récursifs peuvent être complexes à mettre en œuvre et à optimiser d'un point de vue implémentation. Des structures latentes moins contraintes pourraient présenter des avantages en terme de facilité d'implémentation [7].

Il est admis que les **transformers sont sur-paramétrés** [1, 2, 3, 4, 5]. Pourtant, le rôle de cette sur-paramétrisation n'est pas bien compris. Nous étudions le rôle des couches multiples dans les modèles de transformers profonds.

Nous interprétons les **couches comme un processus itératif** où les représentations contextualisées des tokens sont progressivement raffinées.

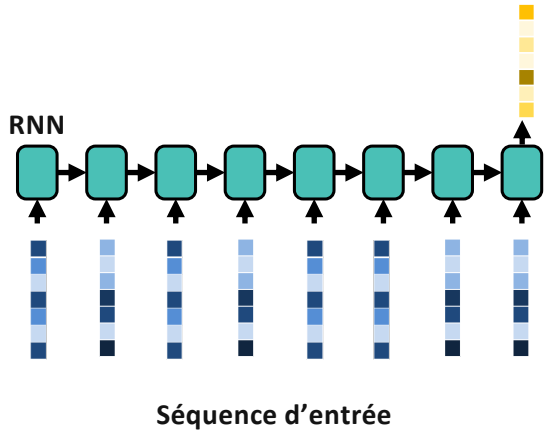
Nous concevons une variante d'ALBERT [6] qui **adapte dynamiquement le nombre de couches pour chaque token de l'entrée**.

[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

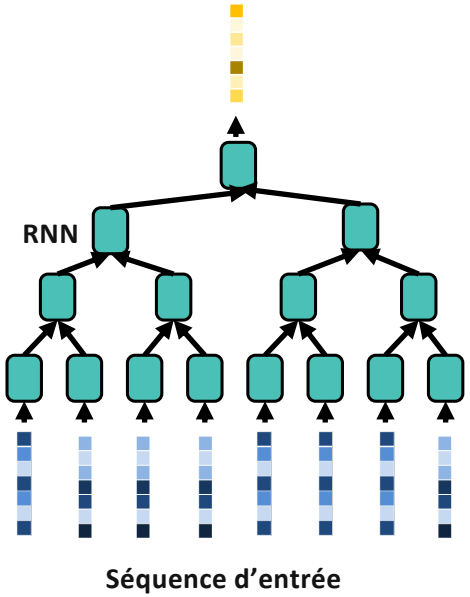
3. Structures latentes dans les modèles de transformers

L'architecture transformers

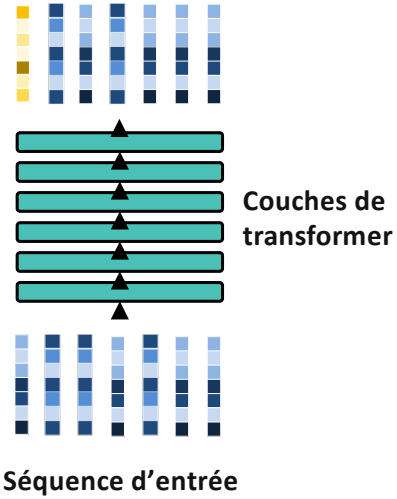
Nous avons utilisé plusieurs types d'architectures de réseaux de neurones : les méthodes **séquentielles, arborescentes** ou encore en **graphes**.



Réseau de neurones séquentiel



Réseau de neurones arborescents



Réseau de neurones en graphes

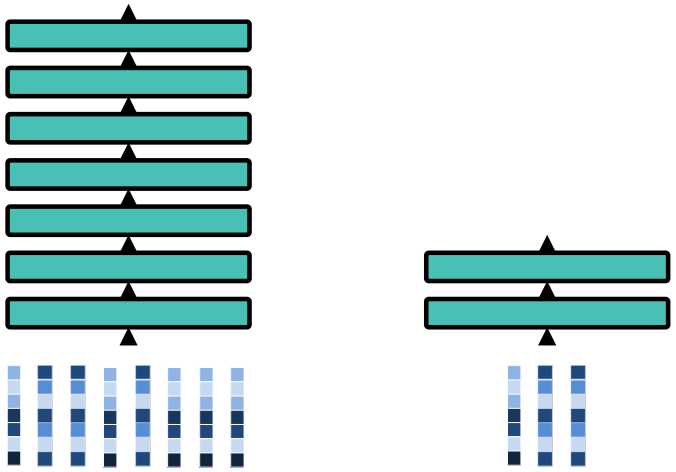
3. Structures latentes dans les modèles de transformers

Méthode

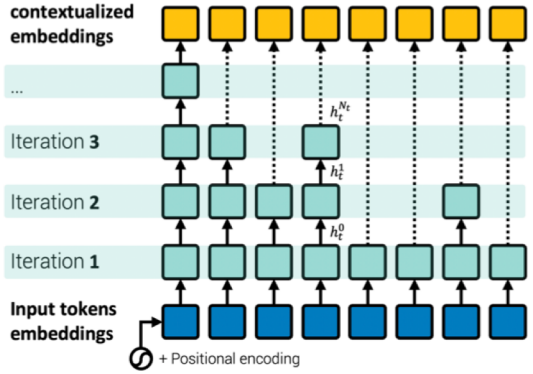
Certains modèles [1, 2, 3] proposent **d'adapter dynamiquement le nombre de couches à la complexité de la phrase.**

De nombreuses couches peuvent être nécessaires pour extraire des informations sémantiques d'un paragraphe complexe complet ...

... mais peut être disproportionné pour traiter une phrase de 5 mots.



Le modèle peut **adapter le nombre de couches appliquées pour chaque token** en utilisant un mécanisme de calcul adaptatif [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. Structures latentes dans les modèles de transformers

Expériences (1/2) Analyse du pré-entrainement

Analyse des itérations : le modèle distribue les itérations pour les tokens cruciaux ou difficiles. [CLS] et [MASK] reçoivent plus d'itérations que les autres tokens alors que [SEP] en reçoit moins.

Point fixe naturel : Par conception, nous encourageons le modèle à converger vers un point fixe. **Tous les tokens atteignent un point fixe en suivant un schéma de convergence distinct.**

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. Nombre moyen d'itérations en fonction des types de tokens pendant le pré-entraînement. Pour chaque modèle, nous rapportons un nombre moyen d'itérations sur notre ensemble de développement, à la fin du pré-entraînement.

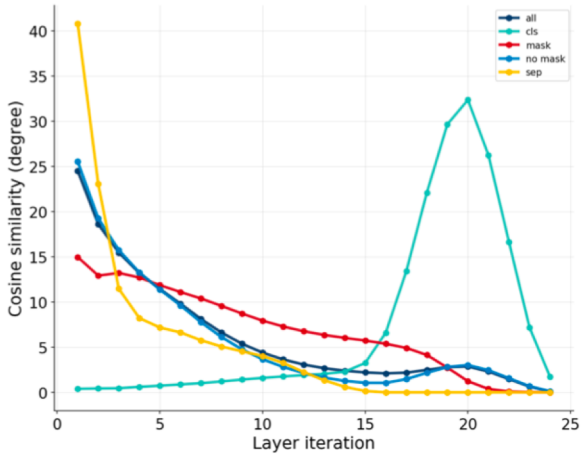


Figure 1. Évolution de la similarité en cosinus entre les états cachés h^n et h^{n+1} de deux itérations consécutives. Nous mesurons les itérations sur notre ensemble de développement, à la fin du pré-entraînement.

3. Structures latentes dans les modèles de transformers

Expériences (2/2) tâches d'applications

Probing tasks: Nous évaluons le modèle sur des sondes linguistiques [2] et nous **comparons le nombre d'itérations effectuées pour chaque token en fonction de son type de dépendance.**

Test de contrôle : Nous évaluons notre configuration sur le benchmark GLUE [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 des itérations selon les types de dépendance des tokens.

	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. Résultats du test GLUE.

[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. Structures latentes dans les modèles de transformers

Conclusion

Les **modèles transformers profonds sont sur-paramétrés**, et la plupart de leurs paramètres sont redondants.

Une meilleure compréhension du mécanisme interne en place dans les modèles de transformers profonds apparaît comme une nécessité pour **réduire la puissance de calcul ou fournir de nouvelles méthodes de régularisation**.

Nos expériences fournissent une nouvelle voie d'interprétation pour le rôle des couches dans les modèles de transformers profonds : Les **couches pourraient être interprétées comme l'itération d'un processus itératif**.

4 Relations entre les propriétés compositionnelles et les structures neuronales

4. Relations entre propriétés compositionnelles et structures neuronales

Problématique et hypothèses de travail

Dans les chapitres précédents, on a proposé différentes architectures de réseaux de neurones avec divers degrés de contraintes. On a évalué pour chacune leur apport pour résoudre des tâches nécessitant d'encoder la sémantique globale de la phrase. On cherche maintenant à raffiner l'évaluation pour quantifier l'impact sur les propriétés de compositionalité.

Les modèles linguistiques peuvent **s'appuyer sur des phénomènes moins systématiques** tels que des biais lexicaux ou des heuristiques superficielles dans les données pour accomplir la tâche [1].

Il est sans aucun doute **possible d'entraîner des modèles de langage efficaces sans propriétés de composition préalables**.

L'évaluation des propriétés de composition des modèles est notoirement difficile. **La création des données est une étape critique.**

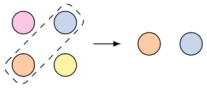
*[1] Tal Linzen, Brian Leonard: **Distinct patterns of syntactic agreement errors in recurrent networks and humans.** CogSci 2018*

4. Relations entre propriétés compositionnelles et structures neuronales

Méthode

Nous introduisons **CobA**, **Co**mpositionality **b**enchmark using **A**rithmetic. Il est facile de construire des exemples arithmétiques en isolant des propriétés spécifiques. Nous entraînons le modèle à évaluer les expressions du domaine. Nous inférons ensuite les expressions à partir de l'ensemble de généralisation qui comprend des partitions pour **Systématicité, Productivité**, **Substitutivité**, et **Localisme**.

Systématicité
Recombinaison de parties connues pour former de nouvelles séquences.



Entrainement

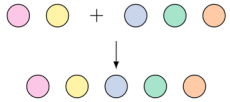
12×4

7×33

Evaluation

12×7

Productivité
Extrapolation à des séquences plus longues.



Entrainement

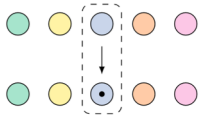
$3 + 4$

$5 + 4$

Evaluation

$4 + 4 + 2 + 5$

Substitutivité
Robustesse à l'introduction de synonymes.



Entrainement

3×4

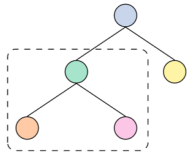
5×4

Evaluation

6×7

7×6

Localisme
Evaluation récursive des petits constituants avant les grands constituants.



Entrainement

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. Relations entre propriétés compositionnelles et structures neuronales

Expériences (1/2) tâches d'évaluation

En général, les en modèles reposant sur des cellules LSTM sont plus performants que les transformers. Pour la productivité et la systématicité, les modèles séquentiels surpassent fortement les modèles utilisant un contexte de longueur fixe. En ce qui concerne les transformers, nous confirmons leur limitation connue en termes de productivité.

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. Évaluation de la compositionnalité.

4. Relations entre propriétés compositionnelles et structures neuronales

Expériences (2/2) Impacte de la longueur des phrases

La productivité est notoirement difficile. Nous décomposons l'ensemble de généralisation de la productivité en fonction du nombre d'opérateurs et du nombre d'exemples d'exposition et traçons l'évolution de la RMSE.

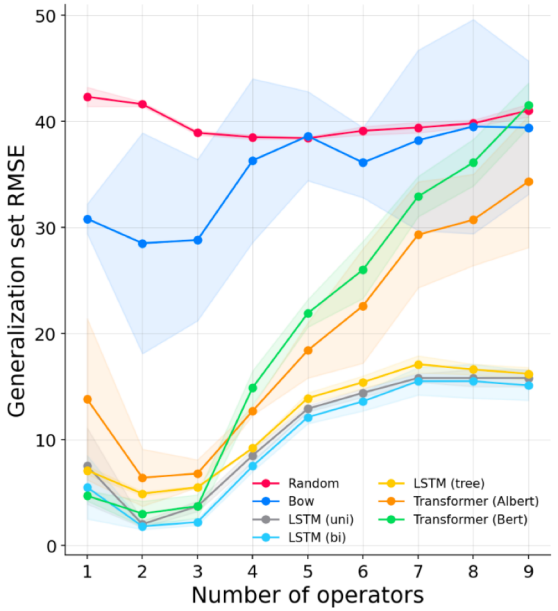


Figure 1. Evolution du RMSE pour la productivité en fonction du **nombre d'opérateurs**.

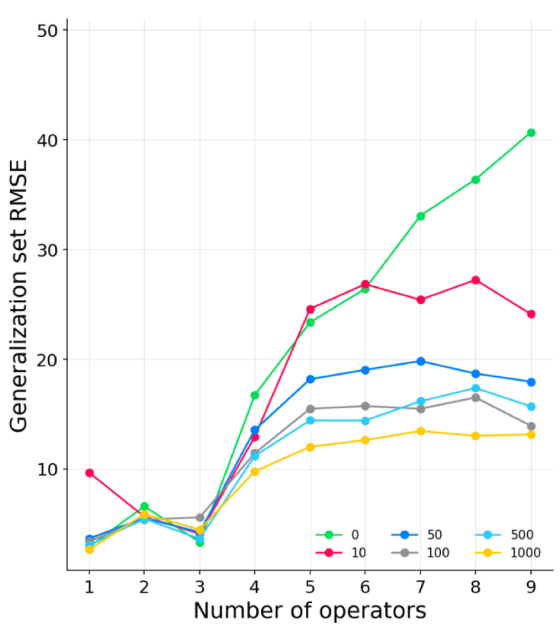


Figure 2. Evolution du RMSE pour la productivité en fonction du **nombre d'exemples d'exposition**.

4. Relations entre propriétés compositionnelles et structures neuronales

Conclusion

En général, les modèles sont **robustes à l'introduction de paraphrases** (substitutivité) et peuvent effectuer une **évaluation récursive des sous-composants** (localisme).

Les **transformers peinent à se généraliser à des séquences plus longues** (productivité) **ou à combiner des parties connues pour former de nouvelles séquences** (systématicité).

Les modèles **peinent avec des expressions complexes, impliquant plus de symboles, une structure plus complexe, ou plus de types d'opérateurs**. Nous améliorons légèrement le degré de compositionnalité en adaptant le nombre de paramètres ou l'exposition aux expressions de généralisation.

Conclusion et travaux futurs

Conclusion

L'introduction d'un **à priori de structure** fort dans les réseaux de neurones permet dans certaines configurations de construire de **meilleures représentations**. En particulier en combinant des structures variées ou en apprenant la structure en même temps que la fonction de composition.

Sur de nombreuses taches, les **transformers**, qui apportent très peu d'hypothèses de structures, continuent d'obtenir les résultats à l'état de l'art. Nous proposons une **interprétation de ces réseaux selon un formalisme de graphe**.

Une **évaluation plus fine des propriétés de compositionalité** des modèles et mettons en lumière la limite des transformers sur certaines propriétés clés.

Dans une seconde partie de la thèse nous avons comparé les approches de modélisation de structure par un **passage à l'échelle des modèles et des données.**

Nous **proposons plusieurs ressources** : des **corpus** d'évaluation de la compositionalité ou des capacité de génération des modèles en français, des modèles pré-entrainés en français, des **modèles** de plongement de phrases à l'état de l'art pré-entrainés en anglais, des **librairies** d'implémentation des réseaux récursifs.

Contributions

Articles de conférences

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*

Logiciels

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

Ressources

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

Questions ?

Annexes

1. Plongements de phrases par combinaison de différentes structures

Evaluation sur le benchmark SentEval

Pour l'expérience 1, Les **résultats détaillés sur le benchmark SentEval.**

Model	Dim	Hrs	MR	CR	SUBJ	MPQA	TREC	MRPC Acc	MRPC F1	SICK-R r	SICK-R ρ	SICK-R MSE
Context sentences prediction												
FastSent	≤ 500	2	70.8	78.4	88.7	80.6	76.8	72.2	80.3	—	—	—
FastSent + AE	≤ 500	2	71.8	76.7	88.8	81.5	80.4	71.2	79.1	—	—	—
Skipthought	4800	336	76.5	80.1	93.6	87.1	92.2	73.0	82.0	85.8	79.2	26.9
Skipthought + LN	4800	672	79.4	83.1	93.7	89.3	—	—	—	85.8	78.8	27.0
Quickthoughts	4800	11	**80.4**	**85.2**	**93.9**	**89.4**	**92.8**	**76.9**	**84.0**	**86.8**	**80.1**	**25.6**
Sentence relations prediction												
InferSent	4096	—	**81.1**	**86.3**	92.4	90.2	88.2	**76.2**	**83.1**	**88.4**	—	—
DisSent Books 5	4096	—	80.2	85.4	93.2	90.2	91.2	76.1	—	84.5	—	—
DisSent Books 8	4096	—	79.8	85.0	**93.4**	**90.5**	**93.0**	76.1	—	85.4	—	—
Pre-trained transformers												
BERT-base [CLS]	768	96	78.7	84.9	94.2	88.2	**91.4**	71.1	—	75.7†	—	—
BERT-base [NLI]	768	96	**83.6**	**89.4**	**94.4**	**89.9**	89.6	**76.0**	—	**84.4**†	—	—
Our models (GloVe & Pretrained Embeddings)												
SEQ, CONST†	4800	41	79.8	82.9	94.6	88.5	90.4	76.4	83.7	86.1	78.9	26.3
DEP, SEQ†	4800	27	79.7	82.2	94.4	88.6	91.0	**77.9**	**84.4**	86.6	79.8	25.5
DEP, CONST†	4800	39	**80.7**	**83.6**	**94.9**	**89.2**	**92.6**	76.8	83.6	**87.0**	**80.3**	**24.8**

Table 1. Résultats détaillés sur SentEval

1. Plongements de phrases par combinaison de différentes structures

Evaluation sur le benchmark SentEval

Pour l'expérience 1, on **compare les différentes méthodes d'apprentissage : complètement supervisé ou auto-supervisé**. Pour l'apprentissage auto-supervisé on apprend un classifier prenant les plongements de phrases comme données d'entrée.

Encoder	r	ρ	MSE
BOW	78,2 (1,1)	67,9 (1.6)	40.0 (1.9)
SEQ (uni-directional)	84.6 (0.4)	78.4 (0.5)	29.2 (0.8)
SEQ (bidirectional)	85.1 (0.4)	78.7 (0.4)	28.5 (0.7)
CONST	85.3 (0.7)	79.3 (0.6)	28.2 (1.4)
DEP-FREE (free-init)	85.8 (0.4)	79.4 (0.7)	27.0 (0.6)
DEP	86.5 (0.4)	80.5 (0.5)	25.9 (0.8)
DEP-FREE (train-init)	87.0 (0.1)	81.3 (0.2)	24.9 (0.3)
BERT-base	87.3 (0.9)	81.5 (1.4)	25.7 (2.3)
BERT-large	89.4 (0.6)	84.7 (1.0)	21.5 (1.3)

Table 1. Apprentissage **supervisé**

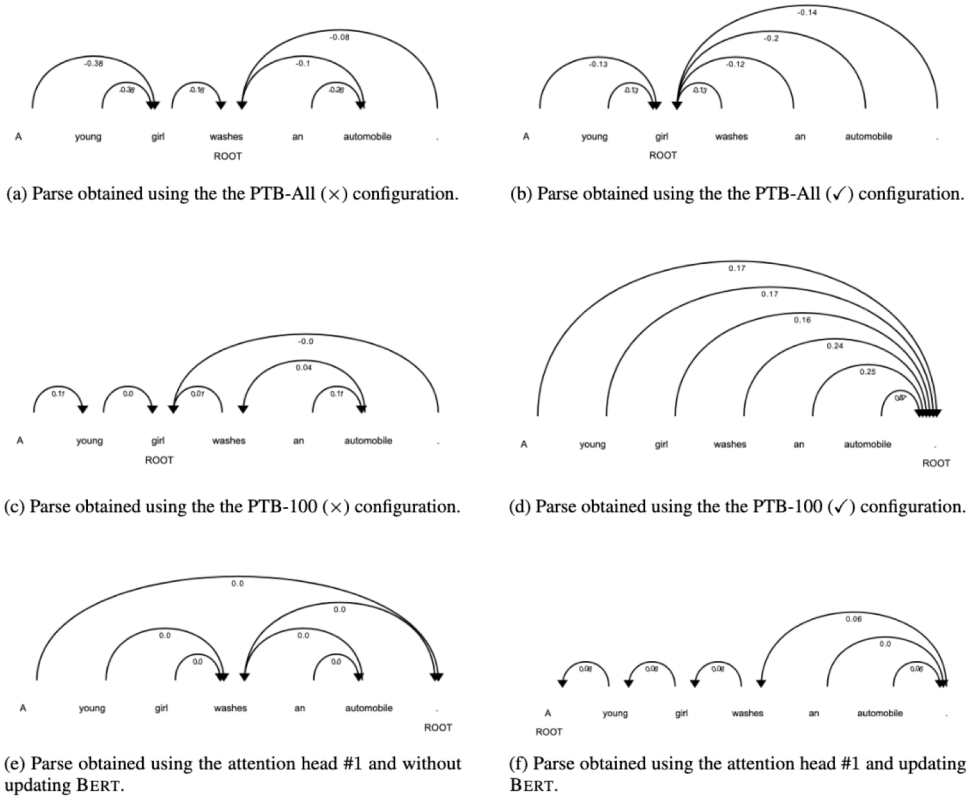
Encoder	r	ρ	MSE
SEQ, SEQ	85.4	77.7	27.6
DEP-FREE, SEQ (free-init)	85.6	77.8	27.3
DEP-FREE, SEQ (train-init)	85.8	78.3	26.9
DEP, SEQ	**86.0**	**78.9**	**26.6**

Table 1. Apprentissage **auto-supervisé**

2. Apprentissage conjoint de la structure et des opérations de composition

Expériences (1/2) évaluation sur des tâches d'applications

Pour l'expérience 2, **Exemple de structures apprise par le modèle sur la tâche SNLI**.



(a) Parse obtained using the the PTB-All (×) configuration.

(b) Parse obtained using the the PTB-All (✓) configuration.

(c) Parse obtained using the the PTB-100 (×) configuration.

(d) Parse obtained using the the PTB-100 (✓) configuration.

(e) Parse obtained using the attention head #1 and without updating BERT.

(f) Parse obtained using the attention head #1 and updating BERT.

Figure 1. Exemple de structure apprise par le modèle sur la tâche SNLI (expérience 2)