Sujet de thèse LIPN/EUR: utilisation des *Vision Transformers* pour le traitement automatique des langues

Encadrants :
— Thierry Charnois, PU
— Joseph Le Roux, MCF

1 Introduction

De nombreuses tâches en traitement automatique des langues (TAL), *e.g.* analyse syntaxique en dépendances, analyse sémantique, analyse discursive, extraction de relations) manipulent des graphes orientés dont les sommets représentent des mots ou des groupes de mots d'un texte et les arcs, les relations (syntaxiques, sémantiques, discursives...) entre ces mots ou ces groupes. Des contraintes additionnelles spécifiques à certaines tâches peuvent être imposées pour assurer la bonne formation de ces graphes, par exemple des contraintes d'arité, la connexité ou l'acyclicité. Dans ce cadre, une analyse peut se réduire à un problème de sous-graphe optimal à partir du graphe complet, c'est-à-dire au graphes où tous les arcs joignant deux mots sont considérés.

Depuis 2017, l'architecture neuronale appelée *Transformer* définie dans [8] a contribué à améliorer les performances des systèmes de TAL en permettant d'extraire des caractéristiques (*features*) contextuelles associées aux mots efficacement quelle que soit la longueur des phrases à analyser. Cette architecture utilise l'attention, c'est à dire une convolution de taille arbitrairement grande, pouvant aller jusqu'à la taille du texte complet, pour calculer des interactions à longue distance entre les éléments d'une phrase. Généralement, plusieurs blocs de *transformers* sont empilés de façon à affiner itérativement ces *features*. Récemment, les *vision transformers* utilisent la même idée mais pour induire ces interactions entre pixels [2]. Ces pixels correspondent à un point de coordonnées (i, j) sur une grille de taille (H, W). En pratique, ces réseaux ne travaillent plutôt au niveau des *patches*, c'est-à-dire des carrés de p pixels.

Dans ce sujet de thèse, nous proposons de modéliser les arcs par des *vision transformers*. Par analogie avec une image, un arc entre mots d'une phrase dans le cas des applications TAL peut s'envisager comme un point de coordonnées (i, j) s'il connecte le $i^{\text{ème}}$ mot

de la phrase au $j^{\text{ième}}$. Si l'on considère une phrase de N mots, on peut représenter le graphe complet par une grille de taille (N, N). Nous voulons explorer dans cette thèse les implications de cette analogie pour les tâches de TAL.

2 Approches existantes et limitations

Dans les travaux récents en analyse pour le TAL, tels que [10] ou [9] pour l'analyse syntaxique, les architectures neuronales sont relativement stables : tout d'abord des réseaux dits *extracteurs de caractérisitique* (comme des réseaux récurrents par exemple) extraient des représentations contextuelles des mots de la phrase qui sont ensuite spécialisées (dans nos exemple, pour les deux rôles syntaxiques de la théorie des dépendances : gouverneur ou adjoint). Puis une fonction biaffine [3] combine les représentations spécialisées pour donner un score à tous les arcs entre mots, c'est-à-dire dans le cas de l'analyse syntaxique à tous les arcs pouvant être sélectionnés pour construire l'arbre d'analyse. En dernière étape, un algorithme combinatoire calcule le sous-graphe optimal à partir des scores d'arcs.

Dans [5], le système réévalue les scores attribués par la fonction biaffine à partir d'un réseau de neurones à convolutions sur les graphes [1] appliqué au graphe complet de la phrase. Expérimentalement, cette approche semble donner de bonnes performances. Il semble donc que réévaluer la solution trouvée en se basant uniquement sur la représentation des mots à l'aune des autres choix qui peuvent être faits pour les autres arcs permet de corriger un certain nombre d'erreurs. Cette approche présente cependant des limitations :

1. seuls les mots sont représentés dans un espace vectoriel latent : les arcs ne sont représentées que par un unique score ;
2. elle est limitée à une tâche seule.

Tout comme les *transformers* peuvent s'envisager comme une généralisation des convolutions, les *vision transformers* peuvent être vus comme des généralisations des réseaux de neurones à convolutions sur les graphes. Ils permettraient ainsi de lever les limitations mentionnées plus haut :

1. chaque arc/pixel est naturellement représenté par un vecteur, et non un simple score : on travaille directement sur la représentation des arcs ;
2. le fait de travailler sur des arcs permet de combiner de des informations utilisables pour différentes tâches (informations communes ou spécifiques).

Enfin de la même façon que les *vision transformers* peuvent être utilisés pour des tâches de classifications d'image en rajoutant un *patch* supplémentaire représentant l'image, on peut représenter la phrase entièrement pour les tâches de classification de phrases, comme l'analyse de sentiment.

3 Verrous

L'un des problèmes connus de *vision transformers* est l'absence de notion de localité, contrairement aux convolutions. En effet, dans cette architecture un arc peut tirer des informations de n'importe quel autre arc du graphe complet. De nombreuses extensions ont été proposées, en hybridant avec des convolutions ou en procédant à une extension progressive de la taille de l'attention [6] mais selon des schémas qui, bien que pertinents pour les images, sont difficilement applicables aux textes. Par exemple, avec la représentation des arcs en grille donnée plus haut, une colonne représente tous les arcs ayant la même destination. Dans le cas où le sous-graphe optimal doit être un arbre (en analyse syntaxique par exemple), c'est une localité importante à exploiter en priorité.

Un autre problème des *vision transformers* est leur grand besoin de données. Il semble nécessaire de développer des méthodes *d'augmentation de données* [7] où l'on génère des données d'entraînement artificielles en modifiant des données réelles. En image, les données artificielles sont générées en appliquant aux images d'origine des opérations telles que des rotations, zoom, modification des couleurs. Il s'agira de comprendre quelles opérations peuvent être appliquées à la représentation en grille du graphe complet pour générer des données pertinentes.

Enfin, ces réseaux de neurones très expressifs ne sont utilisés que pour calculer des caractéristiques, et le sous-graphe optimal est souvent calculé par un simple objectif linéaire. On pourrait envisager d'utiliser ces réseaux pour calculer directement le score des sous-graphes. Évidemment il n'y a plus d'algorithme tractable dans ce cas pour retourner le sous-graphe optimal, et il faut se tourner vers des méthodes approchées. On peut penser aux méthodes d'échantillonnage [4] ou aux méthodes variationnelles [9].

4 Plan de travail

Puisque nous souhaitons nous inspirer de [5] mais en remplaçant la convolution sur les arêtes adjacentes par un transformer sur une grille où chaque point (i, j) de cette grille représente un arc entre un mot i et un mot j, cela revient donc en pratique à un *vision transformer*. Nous découpons a priori le travail de thèse comme suit :

1. état de l'art sur les *nombreux* articles développant les *transformers* et les *vision transformers*, notamment les études portant sur la localité et la génération d'exemples artificiels ;
2. développement de cette architecture en deux temps : d'abord dans le cadre d'une seule tâche (analyse syntaxique, extraction de relations) puis dans le cadre de l'apprentissage multi-tâche ;
3. études des contraintes de localités pertinentes pour les problèmes d'analyse considérés au point précedent et développement de méthodes d'augmentation de données ;
4. études des méthodes approchées pour utiliser le *vision transformer* pour retourner directement la structure optimale.

Selon le profil du candidat, les points 3 et 4 pourront être plus ou moins approfondis.

Références

[1] Joan Bruna et al. “Spectral networks and locally connected networks on graphs”. English (US). In : *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*. 2014.

[2] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale”. In : *International Conference on Learning Representations*. 2021. URL : https://openreview.net/forum?id=YicbFdNTTy.

[3] Timothy Dozat et Christopher D. Manning. “Deep Biaffine Attention for Neural Dependency Parsing”. In : *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL : https://openreview.net/forum?id=Hk95PK9le.

[4] Will Grathwohl et al. “Oops I Took A Gradient : Scalable Sampling for Discrete Distributions”. In : *Proceedings of the 38th International Conference on Machine Learning*. Sous la dir. de Marina Meila et Tong Zhang. T. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, p. 3831-3841. URL : https://proceedings.mlr.press/v139/grathwohl21a.html.

[5] Tao Ji, Yuanbin Wu et Man Lan. “Graph-based Dependency Parsing with Graph Neural Networks”. In : *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy : Association for Computational Linguistics, juill. 2019, p. 2475-2485. DOI : 10.18653/v1/P19-1237. URL : https://www.aclweb.org/anthology/P19-1237.

[6] Ze Liu et al. “Swin Transformer : Hierarchical Vision Transformer using Shifted Windows”. In : *CoRR* abs/2103.14030 (2021). arXiv : 2103.14030. URL : https://arxiv.org/abs/2103.14030.

[7] Connor Shorten et Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. In : *J. Big Data* 6 (2019), p. 60. DOI : 10.1186/s40537-019-0197-0. URL : https://doi.org/10.1186/s40537-019-0197-0.

[8] Ashish Vaswani et al. “Attention is All you Need”. In : *Advances in Neural Information Processing Systems*. Sous la dir. d’I. Guyon et al. T. 30. Curran Associates, Inc., 2017. URL : https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[9] Xinyu Wang et Kewei Tu. “Second-Order Neural Dependency Parsing with Message Passing and End-to-End Training”. In : *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China : Association for Computational Linguistics, déc. 2020, p. 93-99. URL : https://aclanthology.org/2020.aacl-main.12.

[10] Yu ZHANG, Zhenghua LI et Min ZHANG. "Efficient Second-Order TreeCRF for Neural Dependency Parsing". In : *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online : Association for Computational Linguistics, juill. 2020, p. 3295-3305. DOI : 10.18653/v1/2020.acl-main.302. URL : https://www.aclweb.org/anthology/2020.acl-main.302.