Étude de l'adaptation des gros modèles de langues par retour visuel

Studiying Adapters Training via Visual Cues

Iskandar Boucharenc, Sahar Ghannay, Christophe Servan, Laure Soulier, Sophie Rosset

English Abstract—Pretrained models have gained popularity lately while their parameters' number have exploded. New techniques have been developped to specialize them, the adapters. Leveraging visualisation tools, we didactically try to explain their efficiency by studiying similarities layer-wise and example-wise between adapters and full fine-tuning.

1 Introduction

L'efficacité des modèles de langue pré-entraînés (PLM) [1] a précipité leur démocratisation et l'utilisation du *fine-tuning* (FT). L'augmentation du nombre de leurs paramètres a motivé le développement de techniques d'entraînement frugales. Parmi celles-ci, les modèles de type *adapters* [7] ont montré leur intérêt. Si le principe est simple au premier abord, la manière dont ils modulent les sorties des PLM n'est pas claire. Pour mieux comprendre cette modulation, nous menons une étude comparative entre l'entraînement de ces *adapters* et le FT. Nous comparons à l'aide de mesures de similarité les différentes couches après entraînement suivant ces deux paradigmes. Puis, pour les comparer plus finement nous étudions l'évolution des représentations en employant des techniques de réduction de dimension. Cette étude donne des pistes pour expliquer l'efficacité des méthodes *adapter* et à la compréhension de l'efficacité du FT.

2 Contexte

Depuis quelques années, plusieurs méthodes d'apprentissage fines ont été conçues pour ne mettre à jour qu'une fraction des paramètres des PLM. Les modèles *adapters* font partie de cette classe [5]. Introduits initialement pour la classification d'images [7], ils ont été utilisés en traitement automatique des langues (TAL) sous la forme des *bottleneck adapters* (BN) [3]. Ces derniers ont montré des résultats encourageants sur les modèles à base de *transformers* [9], notamment BERT [1]. Techniquement, ils consistent à insérer des couches linéaires au sein des couches des *transformers*. La phase d'entraînement consiste à figer les poids du PLM, excepté les couches ajoutées. Le modèle final comporte moins de paramètres à mettre à jour lors de l'entraînement. Les BN ne sont, en revanche, pas des modèles entiers par eux-même, ils s'associent au modèle original pour l'inférence.

Ainsi, la tâche apprise par les BN n'est pas totalement explicite, ils modulent les sorties des couches profondes du PLM pour inférer le résultat voulu. La manière dont cette modulation s'opère n'est pas claire. Nous souhaitons répondre aux questions suivantes : les modèles d'adaptation apprennent-ils une représentation de la tâche finale ? Ou utilisent-ils un savoir déjà présent dans le modèle et comment ?

Dans ce travail nous analysons les hypothèses suivantes : les BN modulent les sorties des couches à la manière d'un FT sur le modèle ; les BN apprennent à détecter des sous-structures du modèle original qui sont naturellement efficaces pour la tâche finale et focalisent l'information sur celles-ci.

3 Méthodes

Nous comparons les paradigmes appliqués au PLM BERT sur les tâches MRPC et STS-B du benchmark GLUE [10]. Une première méthode permet d'observer grossièrement la similarité des couches du modèle sur l'ensemble des exemples. Puis, en utilisant les techniques de réduction de dimension, nous étudions l'évolution des représentations des *tokens* `[CLS]` et analysons leurs caractéristiques pour avoir une comparaison plus fine entre les modèles FT et BN.

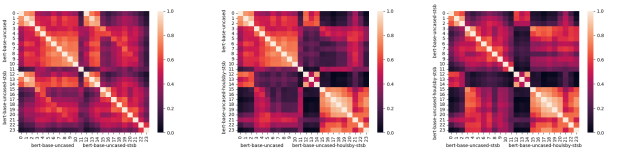
La méthode *Centered Kernel Alignement* (CKA) [4] est une mesure de similarité qui a l'avantage d'être invariante aux homotéthies et aux transformations orthogonales. Le CKA se construit en utilisant le critère d'indépendance d'Hilbert-Schimdt (HSIC) [2].

Les coefficients CKA peuvent alors être visualisés par une carte thermique (*heatmap*) et mettre en exergue les couches les plus similaires. Nous souhaitons déterminer si les méthodes BN, lors de leur apprentissage,

- *Iskandar Boucharenc: LISN**
- *Sahar Ghannay: LISN**
- *Christophe Servan: LISN*, Qwant*
- *Laure Soulier: ISIR***
- *Sophie Rosset: LISN**

**E-mail: {prénom.nom}@lisn.upsaclay.fr.*
***E-mail: laure.soulier@isir.upmc.fr.*

deviennent similaires aux couches du PLM ou s'en éloignent. Nous comparons les cadre BN et FT avec le PLM sans entraînement pour comparaison témoin. Pour analyser les coefficients, nous analysons aussi la progression de leurs quartiles et de leurs valeurs maximales.



(a) Sans entraînement et fine-tuned (b) Sans entraînement et *bottleneck* (c) fine-tuned et *bottleneck adapter*
adapter

Fig. 1: Comparaison des similarités deux à deux sur les données STS-B

L'un des défis pour la visualisation des gros modèles en apprentissage automatique est leur nombre de paramètres ($\simeq 110 \times 10^6$ pour BERT-base). Ainsi les techniques de réduction de dimension apparaissent comme un choix naturel pour visualiser les représentations de mots dans ces modèles.
Nous considérons l'algorithme *Uniform Manifold Approximation and Projection for Dimension Reduction* (UMAP) [6], [8]. En supposant que les données sont réparties uniformément sur une variété différentielle, munie d'une métrique riemannienne localement constante. L'objectif de cette réduction est de conserver les propriétés topologiques de l'espace de départ.
Pour comparer les projections, nous calculons d'abord l'évolution des distances entre les représentations des différents modèles. Pour pousser cette analyse plus loin, nous regardons aussi l'évolution des *clusters* obtenus avec un KNN, notamment l'évolution du ratio de points de chaque modèle dans ces *clusters*.

4 Résultats

En analysant les matrices de coefficients CKA nous observons plusieurs phénomènes.
Premièrement, en comparant le modèle figé (sans entraînement) nous retrouvons des résultats déjà observés par [4]. La similarité entre le PLM et le FT tend à s'effacer progressivement dans les couches passant d'un coefficient CKA de 1 à $0,2$. L'hypothèse généralement posée est que les premières couches représentent des caractéristiques linguistiques et sémantiques générales. Ce résultat se retrouve sur les deux tâches considérées. Concernant les couches d'attention des *adapters*, le constat est semblable mais doit être tempéré avec le fait que nous comparons les représentation des tokens à la sortie des couches d'attention figées. L'effacement en étant encore plus progressif montre que la modulation apportée par les BN est très légère dans les premières couches.
De plus, en étudiant les distributions des coefficients par colonne puis par lignes nous renforçons notre première observation (la similarité diminue plus on

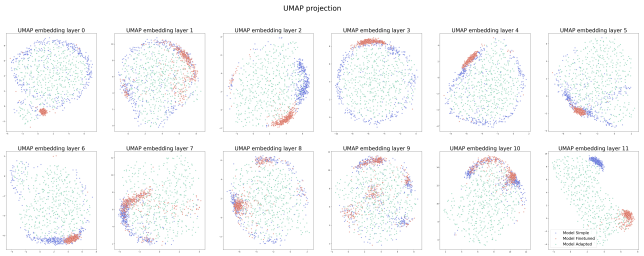


Fig. 2: Projection UMAP les données MRPC, entraînée avec les couches BN (vert), appliquée au PLM (bleu) et FT (rouge)

progresse dans les couches). Nous pouvons ajouter que la comparaison deux à deux des paradigmes montre que la similarité entre les BN et le FT tend plutôt à se conserver dans les couches d'une part et à augmenter dans la dernière. Ainsi une hypothèse naïve que l'on apporte est que les BN conservent les informations utiles et éliminent progressivement les plus superflues à la résolution de la tâche.
Enfin, de manière à moduler nos observations, notons que la similarité CKA est relativement basse, montrant que s'il y a une similarité de l'information celle-ci n'est pas totalement semblable.
Nos résultats avec UMAP sont plus mitigés. Si l'on regarde les sorties des couches d'attention seulement, les représentations semblent rester très proches du modèle de base. Nous observons cependant une plus grande difficulté à séparer les représentations issues des BN et du FT. À l'inverse, regarder les sorties des couches BN donne des représentations complètement différentes pour les couches BN. Une troisième option que nous expérimentons est d'apprendre les réductions sur les couches BN puis les appliquer aux deux autres modèles. Les visualisations laissent alors apparaître un continuum entre les trois modèles tout en séparant le FT et le modèle figé. Ainsi cela nous laisse poser des conjectures proches des précédentes : l'information capturée par les BN emprunte un peu au fine-tuning et conserve des caractéristiques du PLM.

5 Conclusion

En utilisant des techniques classiques en visualisation des données, nous avons étudié sur deux niveaux (couches et exemples) les similarités entre deux paradigmes modernes d'entraînement des modèles (FT et BN). Nous montrons que les méthodes d'adaptation tendent à conserver une similarité à travers les couches du PLM. En outre, en observant l'évolution des exemples après projection sur le plan après réduction des dimensions, nous remarquons que les vecteurs sont plus difficiles à distinguer entre le FT et les BN qu'avec le PLM resté intact. Dans de futures travaux, il s'agira d'expliquer d'une part, pourquoi les coefficients semblent différents sur les tâches considérées alors qu'elles sont proches et que leur corrélation par rang est élevée. D'autre part quelle information du PLM conserve les *adapters*.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[2] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In S. Jain, H. U. Simon, and E. Tomita, editors, *Algorithmic Learning Theory*, pages 63–77, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[3] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for NLP. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019.

[4] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 09–15 Jun 2019.

[5] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9), jan 2023.

[6] L. McInnes, J. Healy, N. Saul, and L. Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

[7] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. Learning multiple visual domains with residual adapters. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 506–516, Red Hook, NY, USA, 2017. Curran Associates Inc.

[8] T. Sainburg, L. McInnes, and T. Q. Gentner. Parametric umap: learning embeddings with deep neural networks for representation and semi-supervised learning. *ArXiv e-prints*, 2020.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[10] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics.