Actes de CORIA-TALN 2023
Paris, France, 5 au 9 juin 2023

Géométrie de l'auto-attention en classification : quand la géométrie remplace l'attention

Loïc Fosse Duc Hau Nguyen Pascale Sébillot Guillaume Gravier
Univ Rennes, CNRS, Inria, INSA Rennes – IRISA, Campus de Beaulieu, 35042 Rennes
loic.fosse@insa-rennes.fr, duc-hau.nguyen@irisa.fr,
pascale.sebillot@irisa.fr, guig@irisa.fr

RÉSUMÉ

Plusieurs études ont mis en évidence l'anisotropie des plongements issus d'un modèle BERT au sein d'un énoncé, c'est-à-dire leur concentration dans une direction donnée, notamment dans une tâche de classification. Dans cet article, nous cherchons à mieux comprendre ce phénomène et comment cette convergence se construit en analysant finement les propriétés géométriques des plongements, des clés et des valeurs dans une couche d'auto-attention. Nous montrons que la direction vers laquelle les plongements s'alignent caractérise la classe d'appartenance de l'énoncé. Nous étudions ensuite le fonctionnement intrinsèque de la couche d'auto-attention et les mécanismes en jeu entre clés et valeurs pour garantir la construction d'une représentation anisotrope. Cette construction se fait de manière progressive lorsque plusieurs couches sont empilées. Elle s'avère également robuste à des contraintes externes sur la distribution des poids d'attention, compensées par le modèle en jouant sur les valeurs et les clés.

ABSTRACT

Geometry of self-attention in classification: when geometry replaces attention

Various studies have highlighted the anisotropy of BERT word embeddings within an utterance, i.e., their concentration in a given direction, especially in a classification task. In this paper, we aim at better understanding this phenomenon and how this convergence is built by analyzing the geometric properties of the word embeddings, keys and values within a self-attention layer. We show that the direction towards which embeddings align themselves characterizes class membership. We then study the intrinsic mechanism of the self-attention layer and the mechanisms at play between keys and values to ensure the construction of an anisotropic representation. This construction is progressive when several layers are stacked. It also proves to be robust to external constraints on the distribution of attention weights, which the model compensates through the values and keys.

MOTS-CLÉS : classification, auto-attention, transformers, bertologie.

KEYWORDS: text classification, self-attention, transformers, bertology.

1 Introduction

Les modèles transformeurs fondés sur le mécanisme d'auto-attention sont au cœur de l'état de l'art en traitement automatique des langues dans de nombreuses tâches. Ce succès est en partie dû à l'existence de modèles pré-entraînés de manière générique comme modèle de langue (causal ou masqué) qui sont ensuite adaptés à une tâche précise (Radford *et al.*, 2018; Devlin *et al.*, 2019;

Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs, pages 137–150.

Conneau & Lample, 2019; Brown *et al.*, 2020).

Afin de mieux appréhender le fonctionnement interne de ces modèles, plusieurs études se sont intéressées aux différentes informations linguistiques portées par les modèles pré-entraînés, typiquement en utilisant les plongements aux différents niveaux d'un transformeur pré-entraîné, pour mesurer leur performance dans différentes tâches linguistiques. On peut par exemple citer (Van Aken *et al.*, 2019; Htut *et al.*, 2019; Jawahar *et al.*, 2019; Kovaleva *et al.*, 2019; Lin *et al.*, 2019) dont les principaux résultats sont résumés par Rogers *et al.* (2020). Ces travaux montrent que les plongements aux différents niveaux encodent des informations linguistiques différentes, de plus en plus riches et complexes au fur et à mesure que l'on monte en abstraction, les dernières couches étant quant à elle très influencées par la tâche utilisée pour le pré-entraînement. Certaines études se sont également penchées sur les propriétés géométriques des plongements aux différents niveaux d'un modèle transformeur (Reif *et al.*, 2019; Ethayarajh, 2019; Hernandez & Andreas, 2021; Fosse *et al.*, 2022). Plusieurs d'entre elles mettent clairement en avant l'anisotropie croissante des plongements des *tokens* d'une même phrase, c'est-à-dire leur concentration dans une direction au travers des différentes couches d'attention (Ethayarajh, 2019; Cai *et al.*, 2021; Fosse *et al.*, 2022). En particulier, Fosse *et al.* (2022) mettent en évidence une concentration forte des plongements dans une même direction lorsqu'un modèle pré-entraîné est adapté pour la classification de texte, tâche pour laquelle la distinction entre les *tokens* n'a au final plus d'importance.

Même si Cai *et al.* (2021) montrent qu'on peut retrouver une isotropie permettant de distinguer les différents *tokens* dans des sous-espaces, cette dernière observation soulève des questions sur la manière dont les couches d'auto-attention construisent cette convergence dans une direction unique, questions au centre de cet article.

De manière intéressante, ces observations sur la géométrie des plongements sont à mettre en parallèle avec les travaux menés sur les poids d'attention (Clark *et al.*, 2019; Bai *et al.*, 2021; Bibal *et al.*, 2022) qui soulèvent de nombreux débats sur l'intérêt de ces poids pour expliquer la décision du modèle (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019; Bibal *et al.*, 2022). De nombreux exemples visuels montrent qu'il est parfois possible d'interpréter *a posteriori* les poids d'attention [1]. Cependant, plusieurs observations troublantes viennent mettre à mal la possibilité d'utiliser ces poids comme explication. D'une part, il a été mis en évidence que la distribution des poids d'attention sur les *tokens* d'une phrase a tendance à être répartie de manière uniforme dans un modèle d'attention associé à un LSTM (Mohankumar *et al.*, 2020; Nguyen *et al.*, 2021), ou, dans des modèles transformeurs, à se concentrer sur des *tokens* peu informatifs comme [CLS] (Clark *et al.*, 2019). Il a également été mis en évidence qu'il est possible de modifier les poids d'attention sans réelle incidence sur la classification (Voita *et al.*, 2019; Jain & Wallace, 2019; Pruthi *et al.*, 2020). Cette propriété est utilisée dans plusieurs travaux pour contraindre les poids d'attention. On peut, par exemple, chercher à les rendre plus parcimonieux, et donc plus plausibles dans l'explication de la décision (Niculae & Blondel, 2017; Mohankumar *et al.*, 2020; Nguyen *et al.*, 2022). Une autre orientation consiste à contraindre les poids d'attention pour guider l'apprentissage ou la génération d'une explication (Nguyen & Nguyen, 2018; McGuire & Tomuro, 2021; Carton *et al.*, 2022; Paranjape *et al.*, 2020).

Ces deux constatations – la convergence intra-phrase des plongements et la tendance à une distribution uniforme des poids d'attention – nous amène dans cet article à nous interroger plus avant sur le rapport entre la géométrie des plongements et la tâche de classification. Au travers d'une analyse théorique et expérimentale du mécanisme d'auto-attention, nous cherchons à déterminer si la convergence des plongements est une propriété intrinsèque des transformeurs et à comprendre comment cette

1. Voir par exemple la figure 1 dans (Clark *et al.*, 2019).

convergence se construit en jouant sur les clés et les valeurs du modèle.

Dans ce travail, nous mettons en évidence que, dans les tâches de classification d'énoncés, l'anisotropie des plongements au sein d'une phrase est liée à la classe : en d'autres termes, les plongements s'alignent dans une direction qui dépend de la classe, ce qui correspond au fonctionnement intrinsèque du modèle. Nous mettons ensuite en évidence le jeu entre clé et valeur qui permet de construire cet alignement des plongements au travers des couches d'auto-attention, y compris lorsque des contraintes externes sont imposées sur la distribution des poids d'attention. Dans ce dernier cas, nous montrons comment le modèle compense les contraintes pour assurer son fonctionnement en alignant dans une direction les plongements.

2 Formalisme, tâches

Nous posons tout d'abord le formalisme du mécanisme d'auto-attention, au cœur des modèles récents pré-entraînés comme modèle de langage. Nous décrivons ensuite succinctement les modèles et les jeux de données utilisés pour les expériences.

2.1 Formalisme et géométrie de l'auto-attention

Le principe fondamental de ces modèles consiste à transformer une séquence de plongements $\mathbf{x} = \{x_1, \ldots, x_n\}$, $x_i \in \mathbb{R}^d$, en une nouvelle séquence $\mathbf{y} = \{y_1, \ldots, y_n\}$, $y_i \in \mathbb{R}^d$, la nouvelle représentation y_i étant obtenue à partir de l'attention a_{ij} qu'il faut porter à chacun des *tokens* ($j \in [1, n]$) de la séquence en entrée pour obtenir la nouvelle représentation du *token* i.

En pratique, chaque *token* x_i est projeté dans 3 espaces distincts, l'espace des clés, celui des valeurs et celui de requêtes. On notera $q_i = x_i\ \mathbf{Q}$ la clé à la position i et $k_i = x_i\ \mathbf{K}$ (resp. $v_i = x_i\ \mathbf{V}$) la clé (resp. la valeur). Les trois matrices $\mathbf{Q}$, $\mathbf{K}$ et $\mathbf{V}$, chacune de dimension $d \times d$, jouent le rôle de projection du plongement d'un *token* vers un nouvel espace de même dimension que l'espace d'entrée et constituent les principaux paramètres du modèle. Pour la position i, l'attention à porter à chacun des *tokens* de la phrase pour modifier x_i est donnée par

$$a_{ij} = \frac{\exp\left(q_i \cdot k_j/\sqrt{d}\right)}{\sum_{l=1}^{n} \exp\left(q_i \cdot k_l/\sqrt{d}\right)} \ . \tag{1}$$

L'ensemble des $a_{ij} \in \mathbb{R}^+$ constitue les poids d'attention et permet de définir la nouvelle représentation du *token* i selon

$$y_i = \sum_{j=1}^{n} a_{ij} v_j \ . \tag{2}$$

Il est intéressant de donner à ce stade une interprétation géométrique de ce mécanisme par rapport aux observations de l'introduction. Chaque y_i étant obtenu par une combinaison convexe des v_j à partir de poids d'attention positifs, si les v_j se concentrent dans une direction, y_i ne peut par définition se retrouver que dans le cône défini par les v_j comme illustré à la figure 1. De manière duale, on note que si l'ensemble des clés k_j se concentrent dans un cône, alors les poids d'attention tendent vers

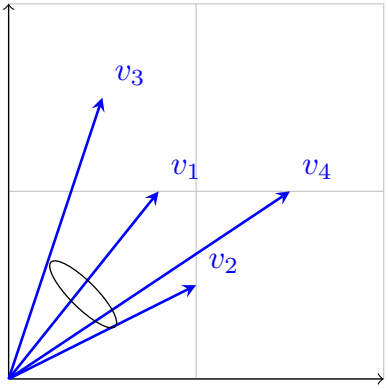


FIGURE 1 – Interprétation géométrique de l'équation 2 : illustration d'un cône défini par l'ensemble des valeurs v_j, $j \in [1, n]$.

une distribution uniforme quelque soit la requête q_i. Enfin, si les requêtes s'alignent, alors les poids d'attention a_{ij} sont identiques pour tous les *tokens* x_i.

2.2 Modèles fondés sur l'auto-attention

Ce mécanisme d'auto-attention correspond à une tête d'attention dans un modèle transformeur (Vaswani *et al.*, 2017), en particulier dans la partie encodeur qui nous intéresse. En pratique, ces modèles s'appuient sur plusieurs couches *transformer* qui modifient progressivement les représentations des *tokens* en entrée du modèle. Chaque couche *transformer* se compose de deux opérations successives : un ensemble de têtes d'attention qui opèrent en parallèle sur la représentation en entrée et dont les sorties sont combinées par concaténation et projection ; une projection point à point de chacun des n vecteurs issus de l'étape précédente. Une connexion résiduelle est utilisée autour de chacune de ces opérations qui sont également chacune suivie d'une normalisation des vecteurs (*layer normalization*). En pratique, chaque tête d'attention opère sur un sous-ensemble des dimensions de l'espace dans lequel les *tokens* sont représentés, la combinaison se limitant alors à une concaténation des représentations issues de chaque tête. Malgré les différentes opérations et l'utilisation de plusieurs têtes, le mécanisme d'attention avec les propriétés géométriques que nous avons défini précédemment reste l'opération centrale au sein d'une couche *transformer*.

La première partie de notre étude s'appuie sur le modèle BERT standard (Devlin *et al.*, 2019) [2], qui comporte 12 couches, chacune composée de 12 têtes d'attention. Chacune des têtes est en charge d'un sous-espace de l'espace complet, soit 12 têtes de dimension 64 pour une dimension totale de 768. Les paramètres du modèle ont été estimés sur un grand volume de données par le biais d'une tâche de modèle de langage avec masque. Lorsque ce modèle est utilisé pour une tâche de classification, une couche de classification est ajoutée et prend en entrée la représentation contextualisée par le modèle BERT du *token* [CLS] de début d'énoncé afin de prédire la classe de l'énoncé. L'ensemble des paramètres du modèle (couche de classification et l'ensemble des couches d'attention) est réestimé par quelques itérations selon un critère standard de minimisation de l'entropie croisée.

Dans la seconde partie de notre étude où nous nous intéressons au fonctionnement intrinsèque des têtes d'attention, nous utilisons une réimplémentation des têtes d'attention dont les paramètres sont estimés

2. Nous utilisons le *checkpoint* `bert-base-uncased` dans le dépôt Huggingface.

modèle	YelpHat	HateXplain	e-SNLI
BERT adapté	93,6	40,1	88,7
auto-attention	91,3	61,1	63,1

TABLE 1 – Performance de classification (en *% d'accuracy*) pour le modèle BERT adapté et pour un modèle avec une couche d'auto-attention.

directement pour la tâche de classification. Le bloc d'attention inclut une couche d'auto-attention à une seule tête (équations 1 et 2) avec une connexion résiduelle et suivi d'une normalisation. Le modèle final de classification utilisé est similaire à celui du paragraphe précédent.

2.3 Tâches et données

Dans cet article, nous nous intéressons au fonctionnement du mécanisme d'attention dans les tâches de classification de texte. Nous exploitons plusieurs jeux de données standards dans le domaine de la classification, en particulier les données liées au *benchmark* ERASER (DeYoung *et al.*, 2020) qui permettent ensuite d'étudier les poids d'attention à l'aune d'une annotation humaine des *tokens* importants pour la classification [3]. Nous utilisons en particulier trois *corpora* en anglais, du plus facile au plus difficile :

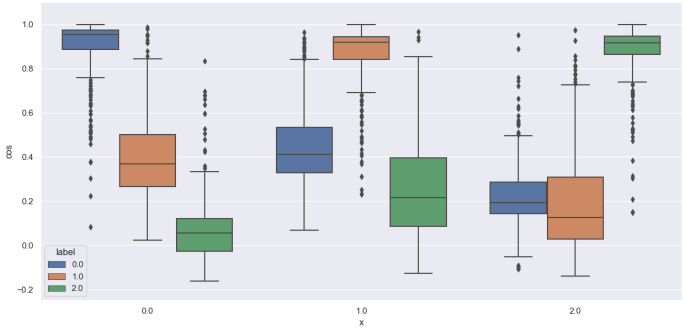
— YelpHat (Sen *et al.*, 2020) vise à la classification en polarité d'avis concernant des restaurants. Le corpus est composé d'environ 3.5k avis pour l'apprentissage et une quantité équivalente pour le test.

— HateXplain (Mathew *et al.*, 2021) a été conçu pour une tâche de classification de messages postés sur des réseaux sociaux en trois catégories : haineux, agressif, neutre. Le corpus est composé de 15k messages pour l'apprentissage et environ 2k pour pour la validation et le test respectivement.

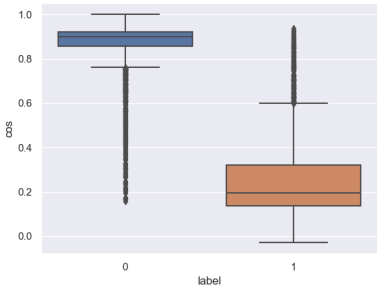
— e-SNLI (Camburu *et al.*, 2018), une extension du corpus SNLI (Bowman *et al.*, 2015), permet une tâche d'inférence linguistique qui consiste à déterminer si deux énoncés constituent une suite logique, sont en opposition ou n'ont aucun rapport. Le corpus contient environ 550k paires de phrase pour l'apprentissage, et environ 10k paires pour la validation et pour le test respectivement. Pour la tâche de classification, les deux énoncés sont concaténés en ajoutant un *token* [SEP] entre eux.

Pour tous ces corpus, la classification est réalisée à partir du plongement contextualisé du *token* [CLS] en utilisant une couche dense de projection. Lors de l'adaptation à la tâche de modèles pré-entrainés, l'ensemble des paramètres sont réestimés sur quelques itérations. L'adaptation à la tâche est effectuée avec les paramètres suivants : graine aléatoire ; optimiseur Adam sur 50 itérations (*epochs*) avec un taux d'apprentissage de 5×10^{-5} pour BERT et de 0.001 pour le second modèle ; stratégie d'arrêt précoce au bout de 5 itérations sans amélioration. Nous rapportons dans le tableau 1 les performances sur les trois corpus, avec le modèle BERT adapté ainsi qu'avec un seul bloc d'attention. On notera simplement les points suivants : le modèle BERT adapté pour HateXplain donne de mauvais résultats, ce qui est cohérent avec plusieurs résultats rapportés sur Internet ; le corpus YelpHat étant facile, l'écart entre BERT pré-entraîné et un modèle avec un seul bloc d'attention est limité.

3. Cette dernière analyse n'entre pas dans le cadre de cette étude mais les données ont été choisies pour pouvoir attaquer cette problématique par la suite.



(a) e-SNLI



(b) YelpHat

FIGURE 2 – Distribution de la similarité cosinus entre le plongement d'un énoncé représentant une classe (r_i) et les plongements du reste des énoncés du corpus (cls_j) en fonction de leur classe pour e-SNLI (2a) et YelpHat (2b).

3 Une convergence en lien avec la classification

Nous étudions tout d'abord la géométrie des plongements issus du modèle BERT adapté à la tâche. Comme dans les études précédentes (Reif *et al.*, 2019; Ethayarajh, 2019; Hernandez & Andreas, 2021; Fosse *et al.*, 2022), nous observons bien évidemment le renforcement de l'anisotropie des plongements au sein d'une même phrase lorsque l'on traverse les couches du modèle, avec une concentration des plongements au sein d'un (hyper)cône dans l'espace des plongements. En effet, si l'on mesure la similarité cosinus moyenne entre les plongements de deux *tokens* au sein d'une même phrase sur la tâche e-SNLI, elle croît de manière *quasi* systématique dans le modèle pré-entraîné, à l'exception notable de la dernière couche. La similarité moyenne passe ainsi progressivement de 0.12 en entrée du modèle à 0.41 en sortie, un maximum (proche de 0.42) étant atteint à la sortie de l'avant-dernière couche. Sur le modèle après adaptation à la tâche, la convergence est nettement plus marquée, passant de 0.08 en entrée à 0.69 en sortie, avec une croissance constante et particulièrement marquée sur les 4 dernières couches où la similarité moyenne passe de 0.34 à 0.69. Un comportement similaire est observé sur d'autres corpus, notamment YelpHat et IMDb (Maas *et al.*, 2011), corroborant ainsi les observations de (Fosse *et al.*, 2022).

Ces observations sont faites énoncé par énoncé et sont valables pour les plongements au sein d'une même phrase. Dans le prolongement, nous nous intéressons donc à l'analyse de ce phénomène d'anisotropie au travers des phrases. Une hypothèse assez naturelle que nous souhaitons vérifier est que, dans le cas d'un modèle BERT adapté à une tâche de classification, la direction prise par les plongements au sein d'un énoncé est caractéristique de la classe de l'énoncé.

Nous vérifions cette hypothèse au travers de l'expérience suivante. Pour chaque classe i dans un corpus, nous choisissons un représentant de cette classe, r_i, représenté par le plongement final du *token* [CLS]. Ce représentant est pris au hasard parmi les énoncés de la classe i qui sont bien classés par le modèle. Nous noterons par la suite cls_j le plongement de [CLS] correspondant au j-ème énoncé du corpus : nous avons donc $r_i = \text{cls}_{i^*}$, i^* étant l'indice d'énoncé du représentant choisi pour la classe i. Pour l'ensemble des énoncés $j \neq i^*$ du corpus, nous mesurons la similarité cosinus entre r_i et cls_j de manière à analyser la distribution des similarités $\cos(r_i, \text{cls}_j)$ en fonction de la classe de l'énoncé j.

phrase	+/-	cos
Last summer I had an appointment to get new tires and had to wait a super long time. I also went in this week for them to fix a minor problem with a tire they put on. They "fixed" it for free, and the very next morning I had the same issue. I called to complain, and the "manager" didn't even apologize ! ! ! So frustrated. Never going back. They seem overpriced, too.	-	
Been coming to cafe rio for awesome fast Mexican food for years. Lived in Utah for a while...just like you Camilla k. I loved the ones in Utah and thought I would give this one a chance. The manager guy, don't know his name, is a total jerk. The food wasn't even warm. It was cold. My wife and I are sitting here right now and I'm so upset that I have to leave this review right now. Just awful service and not even good food anymore. Gradually getting worse. I'm going to costa vida from now on.	-	0,96
SO GOOD ! ! ! ! ! ! ! The only roll I got that wasn't good was a lobster roll. It just had no flavor. Everything else I had was AMAZING ! ! ! Now that I'm done raving about the food, I do have two complaints. 1) The hostess wasn't super friendly or anything. She was really hard to understand and made no effort to speak more clearly so we'd know what she was talking about. 2) There's no where near enough tables. The place is an okay size but there's probably only like 10 tables ? Maybe I'm remembering wrong. There's also no sushi bar, which I don't care about, but some people do.	+	0,93
I am a huge steak person. I live in LA and I've been to every fancy steakhouse in LA, most of the well known steakhouses in NY and Vegas. and Rig may not be the best steakhouse I've ever been to, but it's up there. What truly impressed me was the value. The steaks and appetizers were as good as Mastro's (both Vegas and LA) yet it cost almost half as much. The service was as good. 've told my wife that we'll be going there every time we go to Vegas.	+	0,2
Expensive Gringo Mexican food. Saving grace is the setting. Wonderful pond with ducks in the middle of the facility. Go for the beauty of it, not for the "Mexican" food. They seem to cater to the Cave Creek tourist "semi cowboy" trade.	-	0,09

TABLE 2 – Exemples d'énoncés positifs (+) et négatifs (-) du corpus YelpHat et de leur distance à un énoncé négatif représentatif (ligne 1).

La figure 2 montre sur la partie gauche les distributions obtenues pour chacune des trois classes de e-SNLI (resp. suite logique, sans rapport, contradiction) et, sur la partie droite, celles obtenues pour la classe négative (0) du corpus YelpHat. Ainsi, pour la figure 2b, on observe nettement que les plongements cls_j de la classe négative présentent très majoritairement une grande similarité avec l'énoncé représentatif de cette classe, tandis que les plongements cls_j pour la classe positive sont presque majoritairement orthogonaux. De manière similaire, on voit sur la figure 2a que la direction des plongements cls_j correspond bien aux classes. On note au passage sur cette dernière figure que si le trois classes sont relativement séparables par rapport à un représentant des classes suite logique ou sans rapport, ces deux dernières classes sont difficilement séparables du point de vue d'un représentant de la classe contradiction.

Ces résultats mettent donc clairement en évidence que la direction vers laquelle le modèle plonge la représentation des *tokens* en entrée est l'élément qui caractérise la classe du document. Ceci traduit le fonctionnement inhérent du modèle pour réaliser sa tâche et constitue donc un comportement souhaitable. Nous avons également vérifié que cette propriété est bien induite par l'adaptation du modèle pré-entraîné : la même analyse des distributions des similarités cosinus effectuée sur le modèle pré-entraîné montre que, dans ce cas, il n'est pas possible de distinguer les classes.

Enfin, nous complétons ces observations par quelques exemples d'énoncés positifs (+) et négatifs (-) issus du corpus YelpHat dans le tableau 2, accompagnés de leur distance (cos) à un énoncé

représentatif de la classe négative (ligne 1) : le premier groupe correspond aux énoncés (resp. - et +) les plus proches de l'énoncé de la ligne 1 ; le second groupe donne l'exemple médian pour la classe + [4] et l'exemple le plus éloigné pour la classe -. On note que la représentation issue du modèle BERT dépasse largement le niveau lexical, avec peu de mots porteurs de polarité en commun dans les énoncés proches du représentant. L'exemple positif le plus proche s'explique en particulier par les aspects négatifs que l'énoncé contient en dépit d'un avis globalement positif.

Ces observations sont à mettre en relation avec deux éléments discutés dans l'introduction. D'une part, elle apporte des éléments au débat sur l'attention comme vecteur d'explicabilité, du moins dans le cas de la classification de texte : si la convergence forte des plongements au sein d'une phrase est bénéfique, voire indispensable à la classification, elle laisse peu d'espoir à l'obtention de quelques *tokens* explicatifs grâce à l'attention. En effet, du fait de la convergence, les requêtes et clés dérivées des plongements convergent également dans une même direction, résultant ainsi en une attention *quasi* uniformément distribuée sur les *tokens*, comme observé dans plusieurs travaux (Mohankumar *et al.*, 2020; Nguyen *et al.*, 2021). D'autre part, les travaux visant à contraindre les poids d'attention pour les rendre parcimonieux – et donc potentiellement plus adaptés à une explication – montrent un succès limité (Niculae & Blondel, 2017; Nguyen & Nguyen, 2018; Mohankumar *et al.*, 2020; McGuire & Tomuro, 2021; Nguyen *et al.*, 2022; Sasaki *et al.*, 2023). La convergence des plongements dans une direction dépendante de la classe éclaire ce débat d'une explication possible, à laquelle nous nous intéressons par la suite : si cette notion de projection relève du fonctionnement intrinsèque des modèles, alors ce dernier compense les contraintes imposées sur l'attention pour permettre la convergence des plongements dans la direction la plus appropriée, et cela d'autant plus facilement que le modèle possède un grand nombre de paramètres.

4 Un jeu entre clés, valeurs et attention

Au vu des résultats précédents, nous nous intéressons à voir si l'alignement des plongements dans une direction représentative de chaque classe dans une tâche de classification est inhérente à la couche d'auto-attention et, le cas échéant, comment l'auto-attention permet de construire cette convergence. Nous nous affranchissons pour cela du modèle pré-entraîné comme modèle de langue et utilisons ici un modèle simplifié à une seule tête d'auto-attention et dont les paramètres peuvent être estimés directement pour la tâche de classification.

4.1 Convergence des clés et des valeurs : un mécanisme inhérent au fonctionnement de la couche d'auto-attention

Nous nous sommes tout d'abord intéressés à comprendre le rôle joué par les clés, les valeurs et, dans une moindre mesure, les requêtes pour construire la convergence des plongements au travers des couches. Nous mesurons en particulier la convergence des clés et des valeurs au sein d'un même énoncé. Pour cela, nous introduisons une métrique de similarité des éléments d'une matrice, largement inspirée de la mesure de conicité de (Ethayarajh, 2019), et donnée par

$$\text{sim}(A) = \frac{2}{n(n-1)} \sum_{i<j} \left[(AA^t) . \frac{1}{\sqrt{\text{diag}(AA^t)\text{diag}(AA^t)^t}} \right] (i,j) \tag{3}$$

4. similarité correspondant à la médiane de la distribution des similarités des exemples positifs dans la figure 2b

		HatexPlain					YelpHat					E-SNLI	
		l=1	**l=2**	**l=3**	**l=4**	**l=5**	**l=1**	**l=2**	**l=3**	**l=4**	**l=5**	**l=1**	**l=2**
	L=1	0.713	-	-	-	-	0.578	-	-	-	-	0.657	-
	L=2	0.691	0.683	-	-	-	0.649	0.556	-	-	-	0.719	0.489
A=K	L=3	0.698	0.688	0.841	-	-	0.597	0.431	0.537	-	-	-	-
	L=4	0.614	0.620	0.748	0.840	-	0.714	0.717	0.702	0.860	-	-	-
	L=5	0.624	0.647	0.777	0.913	0.973	0.584	0.542	0.761	0.816	0.959	-	-
	L=1	0.542	-	-	-	-	0.372	-	-	-	-	0.524	-
	L=2	0.510	0.740	-	-	-	0.409	0.511	-	-	-	0.182	0.746
A=V	L=3	0.605	0.688	0.88	-	-	0.461	0.371	0.494	-	-	-	-
	L=4	0.592	0.561	0.785	0.931	-	0.429	0.613	0.803	0.904	-	-	-
	L=5	0.606	0.673	0.858	0.958	0.992	0.417	0.624	0.780	0.9023	0.972	-	-

TABLE 3 – Mesure de la moyenne des similarité intra-énoncé des clés (K) et des valeurs (V). Les lignes (L) représentent le nombre de couches du modèle, les colonnes (l) représentent la couche à laquelle les mesures sont effectuées.

où $A \in \mathbb{R}^{n \times d}$ est une matrice regroupant un ensemble de n clés ou de n valeurs de dimension d, $\text{diag}(A)$ désignant la diagonale de la matrice A. Une valeur de 1 correspond à une matrice A dont les lignes sont toutes co-linéaires, 0 correspondant à des vecteurs lignes orthogonaux, et -1 à des vecteurs lignes alignés dans des directions opposés [5]. En d'autres termes, $\text{sim}(A)$ mesure à quelle point les lignes de la matrice se concentrent dans une direction, mesurant ainsi l'étroitesse du cône défini par les clés ou les valeurs au sein d'une phrase.

Pour chacun des trois corpus e-SNLI, YelpHat et HateXplain, nous avons entraîné des modèles simplifiés à une tête d'auto-attention (plus connexion résiduelle et normalisation – *cf*. section 2.2) avec un nombre de couches variant de 1 à 5 pour YelpHat et HateXplain et, pour des raisons de temps de calcul, de 1 à 2 pour e-SNLI. Nous mesurons pour chacun de ces modèles et au niveau de chacune des couches la valeur moyenne de la similarité des clés et des valeurs donnée par l'équation 3 au sein d'une phrase. La moyenne est effectuée sur 1 000 énoncés choisis aléatoirement dans les données de test en respectant l'équilibre des classes. L'ensemble de ces résultats est donné dans le tableau 3 où les lignes (L) indiquent le nombre de couches du modèle, chaque colonne (l) correspondant à la mesure de similarité prise au niveau de la couche l. Le premier ensemble de valeurs correspond à la similarité moyenne des clés au sein d'une phrase ($A = K$), le second à celle des valeurs ($A = V$).

Les résultats montrent clairement que la convergence se construit bien au niveau de la couche d'auto-attention et conjointement au niveau des clés et des valeurs, avec des clés et des valeurs fortement similaires entre elles au sein d'une phrase sur la dernière couche des modèles. On note aussi qu'elle se construit progressivement, les modèles avec plus de couches permettant une convergence plus forte. Rappelons que : (a) les clés impactent en premier lieu l'attention, des clés rassemblées dans un cône étroit résultant dans une attention distribuée sur l'ensemble des *tokens* de l'énoncé ; (b) les valeurs impactent les plongements en sortie qui sont définis comme une combinaison convexe de l'ensemble des valeurs.

En résumé, dans sa quête d'alignement des plongements dans une direction correspond à une classe, la couche d'auto-attention s'appuie naturellement sur les valeurs pour assurer cette convergence. Hormis pour e-SNLI, cela s'accompagne d'une convergence des clés qui explique les observations de plusieurs études sur la distribution uniforme de l'attention.

5. Par construction des clés et des valeurs, ce cas ne peut pas arriver en théorie avec des matrices de clés ou de valeurs.

4.2 Une tendance à compenser la régularisation de l'attention

Les observations précédentes nous amènent à nous interroger sur le contrôle de l'attention – par exemple pour offrir des capacités d'explication en forçant la parcimonie ou dans un cadre d'apprentissage guidé dans le cadre des approches en *rationalized learning* – et sur son impact sur la convergence des plongements *via* celle des clés et des valeurs.

Ce que nous avons observé dans les section précédentes montre que la direction prise par le plongement du *token* [CLS] est discriminante pour la tâche de classification. Cette direction est portée par les valeurs v_j pour lesquels le poids d'attention a_{0j} (0 étant l'indice du *token* [CLS]) est grand devant les autres. Dans la pratique, nous avons vu que les poids d'attention ont tendance à être uniformes, le *token* [CLS] étant alors le barycentre des valeurs v_j. En régularisant l'attention pour la rendre plus parcimonieuse, c'est-à-dire en supprimant l'uniformité des poids d'attention, nous changeons cependant la direction prise par le plongement du vecteur [CLS]. Or des études antérieurs montrent que cette opération ne change en rien la performance en classification du modèle, ce qui semble indiquer à son tour que cette régularisation n'empêche pas la convergence des plongements. Notre hypothèse est que pour compenser les contraintes imposées aux poids d'attention tout en garantissant la convergence, le modèle joue sur les valeurs en en renforçant la convergence dans un cône étroit pour une phrase donnée ; ainsi, toute modification de l'attention aura un impact très limité sur le plongement résultant de la combinaison des valeurs. Une alternative réside dans l'orthogonalisation des clés afin de concentrer sur quelques *tokens* la réponse à une requête donnée.

Pour vérifier cette hypothèse, nous reprenons les expériences de régularisation introduites par Nguyen *et al.* (2022) où la fonction de coût à minimiser à l'apprentissage inclut une minimisation de l'entropie des poids d'attention en plus de la minimisation de l'erreur de classification. Un paramètre λ permet de contrôler l'importance de la régularisation des poids d'attention par le critère d'entropie : plus λ est élevé, plus les poids d'attention se concentrent sur un petit nombre de *tokens*. Nous appliquons ici cette approche sur un modèle avec un seul bloc d'auto-attention.

Le tableau 4 rapporte la similarité moyenne des clés et des valeurs ainsi que l'entropie moyenne des poids d'attention pour différentes valeurs de λ. Ces mesures mettent clairement en évidence que lorsque la régularisation prend de l'importance, la concentration des poids d'attention (mesurée par l'entropie) augmente et l'alignement dans un cône des valeurs est renforcé pour compenser et garantir la convergence des plongements finaux dans une direction dépendante de la classe. Parallèlement, ces résultats réfutent l'hypothèse d'orthogonalisation des clés pour garantir la parcimonie de l'attention. Nous avons observé que le modèle joue avant tout sur la norme des clés pour rendre l'attention parcimonieuse plutôt que sur la direction. Cette dernière remarque illustre remarquablement la capacité d'une couche d'auto-attention à tirer parti de ces différents paramètres pour compenser les contraintes qui lui sont imposées et permettre de remplir sa fonction première d'alignement des plongements dans une direction dépendante de la classe.

5 Discussion

Les résultats présentés dans cette étude éclairent plusieurs points sur le fonctionnement des modèles transformeurs et, en particulier, le fonctionnement de la couche d'auto-attention dans les tâches de classification d'énoncé. L'ensemble des observations indique que la concentration des plongements d'un énoncé dans une direction, phénomène observé dans plusieurs études, est inhérente au fonction-

		$\lambda = 0$	$\lambda = 0.0001$	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$
Hatexplain	*H*	0.988	0.31	0.111	0.021	0.01
	Key	0.745	0.669	0.687	0.732	0.727
	Value	**0.578**	**0.637**	**0.782**	**0.828**	**0.772**
E-SNLI	*H*	0.999	0.576	0.64	0.000	0.000
	Key	0.674	0.737	0.795	0.833	0.707
	Value	**0.507**	**0.45**	**0.526**	**0.868**	**0.925**
YelpHat	*H*	0.47	0.48	0.57	0.49	0.07
	Key	0.631	0.657	0.634	0.607	0.672
	Value	**0.303**	**0.391**	**0.375**	**0.366**	**0.483**

TABLE 4 – Entropie moyenne des poids d'attention (H) et similarité intra-énoncé moyenne des clés (K) et des valeurs (V) pour un modèle à 1 couche en fonction de la régularisation des poids d'attention.

nement du mécanisme d'auto-attention qui, pour la tâche de classification, cherche à « envoyer » la représentation de l'énoncé dans une direction qui dépend de la classe. Pour ce faire, le modèle utilise tous les degrés de liberté qui lui sont offerts en mettant en œuvre un jeu entre clés et valeurs : il en résulte une concentration dans une direction des clés et des valeurs. Ces dernières sont en première ligne pour garantir l'alignement des plongements, le modèle compensant les éventuelles contraintes sur les clés (ou sur l'attention) grâce aux valeurs et en jouant sur la norme des clés.

Par certains côtés, ces observations éclairent le débat sur l'utilisation de l'auto-attention pour expliquer une décision en mettant en avant les *tokens* importants dans la décision. Dans un contexte de classification d'énoncé, cette approche semble vouée à l'échec puisque le modèle cherchera forcément à construire sa convergence des plongements vers un cône donné, résultant inexorablement en une attention uniformément distribuée. La construction progressive de la convergence au travers des couches montre également que si une attention doit être utilisée pour expliquer une décision, elle est à considérer dans les premières couches du modèle avant que la convergence ne devienne trop forte.

Enfin, cet éclairage soulève de nombreuses questions sur la manière dont le modèle définit les directions correspondant à chacune des classes. Par exemple, est-ce que l'on a un comportement similaire pour les tâches d'étiquetage avec une direction par étiquette possible ? Est-ce qu'il existe un lien entre la dimension minimale/optimale de l'espace de plongement et le nombre de classe. Si l'on reste dans un contexte de classification d'énoncé, on peut également se demander si certains *tokens* sont responsables de la convergence dans une direction ou une autre, ce qui reviendrait à dire que, *in fine*, le modèle apprend les mots-clés, éventuellement dans leur contexte, qui permettent une bonne performance en classification. Se pose alors la question de comment mettre en évidence de tels mots-clés ou, de manière plus générale, celle de l'exploitation des propriétés géométriques que nous avons mises en évidence pour assurer une explication de la décision de classification.

Références

BAI B., LIANG J., ZHANG G., LI H., BAI K. & WANG F. (2021). Why attentions may not be interpretable ? In *27th Conference on Knowledge Discovery & Data Mining*, p. 25–34.

BIBAL A., CARDON R., ALFTER D., WILKENS R., WANG X., FRANÇOIS T. & WATRIN P. (2022). Is attention explanation ? An introduction to the debate. In *60th Annual Meeting of the Association for Computational Linguistics*, p. 3889–3900.

BOWMAN S. R., ANGELI G., POTTS C. & MANNING C. D. (2015). A large annotated corpus for learning natural language inference. In *2015 Conference on Empirical Methods in Natural Language Processing*, p. 632–642.

BROWN T., MANN B., RYDER N., SUBBIAH M., KAPLAN J. D., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A. *et al.* (2020). Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, p. 1877–1901.

CAI X., HUANG J., BIAN Y. & CHURCH K. (2021). Isotropy in the contextual embedding space : Clusters and manifolds. In *9th International Conference on Learning Representations*.

CAMBURU O.-M., ROCKTÄSCHEL T., LUKASIEWICZ T. & BLUNSOM P. (2018). e-SNLI : Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems*, volume 31, p. 9539–9549.

CARTON S., KANORIA S. & TAN C. (2022). What to learn, and how : Toward effective learning from rationales. In *Findings of the Association for Computational Linguistics*, p. 1075–1088.

CLARK K., KHANDELWAL U., LEVY O. & MANNING C. D. (2019). What does BERT look at ? An analysis of BERT's attention. In *2019 ACL Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 276–286.

CONNEAU A. & LAMPLE G. (2019). Cross-lingual language model pretraining. In *Advances in neural information processing systems*, volume 32, p. 7027–7037.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *17th Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 4171–4186.

DEYOUNG J., JAIN S., RAJANI N. F., LEHMAN E., XIONG C., SOCHER R. & WALLACE B. C. (2020). ERASER : A benchmark to evaluate rationalized nlp models. In *Annual Meeting Association for Computational Linguistics*, p. 4443–4458.

ETHAYARAJH K. (2019). How contextual are contextualized word representations ? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *2019 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, p. 55–65.

FOSSE L., NGUYEN D., SÉBILLOT P. & GRAVIER G. (2022). Une étude statistique des plongements dans les modèles transformers pour le français. In *29th Conference Traitement Automatique des Langues Naturelles*, p. 247–256.

HERNANDEZ E. & ANDREAS J. (2021). The low-dimensional linear geometry of contextualized word representations. In *25th Conference on Computational Natural Language Learning*, p. 82–93.

HTUT P. M., PHANG J., BORDIA S. & BOWMAN S. R. (2019). Do attention heads in BERT track syntactic dependencies ? Unpublished arXiv preprint 1911.12246.

JAIN S. & WALLACE B. C. (2019). Attention is not explanation. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, volume 1, p. 3543–3556.

JAWAHAR G., SAGOT B. & SEDDAH D. (2019). What does BERT learn about the structure of language ? In *Annual Meeting of the Association for Computational Linguistics*, p. 3651–3657.

KOVALEVA O., ROMANOV A., ROGERS A. & RUMSHISKY A. (2019). Revealing the dark secrets of BERT. In *2019 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, p. 4356–4365.

LIN Y., TAN Y. C. & FRANK R. (2019). Open Sesame : Getting inside BERT's linguistic knowledge. In *2019 ACL Workshop BlackboxNLP : Analyzing and Interpreting Neural Networks for NLP*, p. 241–253.

MAAS A. L., DALY R. E., PHAM P. T., HUANG D., NG A. Y. & POTTS C. (2011). Learning word vectors for sentiment analysis. In *49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, p. 142–150.

MATHEW B., SAHA P., YIMAM S. M., BIEMANN C., GOYAL P. & MUKHERJEE A. (2021). HateXplain : A benchmark dataset for explainable hate speech detection. In *35th AAAI Conference on Artificial Intelligence*, p. 14867–14875.

MCGUIRE E. S. & TOMURO N. (2021). Sentiment analysis with cognitive attention supervision. In *34th Canadian Conference on Artificial Intelligence*.

MOHANKUMAR A. K., NEMA P., NARASIMHAN S., KHAPRA M. M., SRINIVASAN B. V. & RAVINDRAN B. (2020). Towards transparent and explainable attention models. In *58th Annual Meeting of the Association for Computational Linguistics*, p. 4206–4216.

NGUYEN D., GRAVIER G. & SÉBILLOT P. (2021). A study of the plausibility of attention between rnn encoders in natural language inference. In *20th IEEE Intl. Conf. on Machine Learning and Applications*, p. 1–7.

NGUYEN D., GRAVIER G. & SÉBILLOT P. (2022). Filtrage et régularisation pour améliorer la plausibilité des poids d'attention dans la tâche d'inférence en langue naturelle. In *29th Conference Traitement Automatique des Langues Naturelles*, p. 95–103.

NGUYEN M. & NGUYEN T. H. (2018). Who is killed by police : Introducing supervised attention for hierarchical LSTMs. In *27th International Conference on Computational Linguistics*, p. 2277–2287.

NICULAE V. & BLONDEL M. (2017). A regularized framework for sparse and structured neural attention. In *31st International Conference on Neural Information Processing Systems*, p. 3340–3350.

PARANJAPE B., JOSHI M., THICKSTUN J., HAJISHIRZI H. & ZETTLEMOYER L. (2020). An information bottleneck approach for controlling conciseness in rationale extraction. In *2020 Conference on Empirical Methods in Natural Language Processing*, p. 1938–1952.

PRUTHI D., GUPTA M., DHINGRA B., NEUBIG G. & LIPTON Z. C. (2020). Learning to deceive with attention-based explanations. In *58th Annual Meeting of the Association for Computational Linguistics*, p. 4782–4793.

RADFORD A., NARASIMHAN K., SALIMANS T. & SUTSKEVER I. (2018). *Improving language understanding by generative pre-training*. Rapport interne, OpenAI.

REIF E., YUAN A., WATTENBERG M., VIEGAS F., COENEN A., PEARCE A. & KIM B. (2019). Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems*, volume 32, p. 8592–8600.

ROGERS A., KOVALEVA O. & RUMSHISKY A. (2020). A primer in bertology : What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, **8**, 842–866.

SASAKI S., HEINZERLING B., SUZUKI J. & INUI K. (2023). Examining the effect of whitening on static and contextualized word embeddings. *Information Processing Management*, **60**(3).

SEN C., HARTVIGSEN T., YIN B., KONG X. & RUNDENSTEINER E. (2020). Human attention maps for text classification : Do humans and neural networks focus on the same words ? In *60th Annual Meeting of the Association for Computational Linguistics*, p. 4596–4608.

VAN AKEN B., WINTER B., LÖSER A. & GERS F. A. (2019). How does BERT answer questions ? a layer-wise analysis of transformer representations. In *28th ACM International Conference on Information and Knowledge Management*, p. 1823–1832.

VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł. & POLOSUKHIN I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, p. 6000–6010.

VOITA E., TALBOT D., MOISEEV F., SENNRICH R. & TITOV I. (2019). Analyzing multi-head self-attention : Specialized heads do the heavy lifting, the rest can be pruned. In *57th Annual Meeting of the Association for Computational Linguistics*, p. 5797–5808.

WIEGREFFE S. & PINTER Y. (2019). Attention is not not explanation. In *2019 Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, p. 11–20.