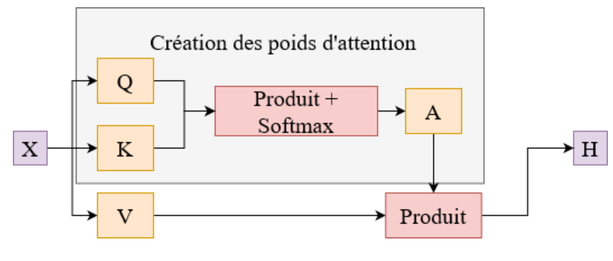
Géométrie de l'auto-attention : quand la géométrie remplace l'attention

Loïc Fosse, Duc-Hau Nguyen, Pascale Sébillot, Guillaume Gravier

Univ Rennes, CNRS, Inria, INSA Rennes – IRISA, Campus de Beaulieu, 35042 Rennes

Objectifs

Les modèles *transformers* sont aujourd'hui à l'état de l'art dans de nombreuses tâches de traitement automatique des langues, notamment à la présence de modèles pré-entrainés sur des tâches génériques (causales ou masquée). Ces modèles partent d'une séquence de plongements et modifient cette dernière *via* le mécanisme d'attention. De nombreuses études ont été portés sur les propriété géométriques des plongements aux différents étages de l'architecture *transformer*, ainsi que à la possibilité d'utiliser les poids d'attention a *posteriori*. Notre objectif est de combiner ces deux points : **utiliser les propriétés géométriques pour comprendre le comportement de l'attention**.

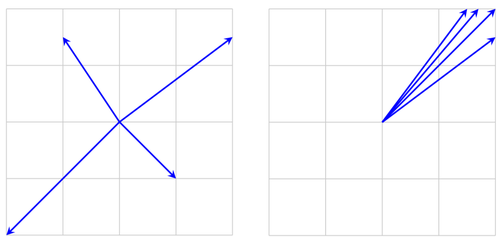


Le mécanisme d'auto-attention pour les *transformers*

1. Des plongements anisotropes

Rappels sur l'anisotropie

L'anisotropie est la concentration des vecteurs dans une direction de l'espace.
Des plongements anisotropes s'organisent dans une structure géométrique semblable à un **cône**.



Un phénomène bien connu

- [Ethayarajh, 2019] montre que l'anisotropie des plongements dans BERT, GPT2 et ELMo
- [Fosse et al, 2022] repartent de cette étude et montrent que l'anisotropie augmente, si on affine les modèles sur une tâche de classification l'anisotropie augmente

Conclusions principales et questions

- **Quel lien entre anisotropie et classification ?**
- **Comment l'auto-attention construit l'anisotropie ?**

2. Formalisme de l'auto-attention

Les modèles étudiés : *transformers* utilisant le **mécanisme d'attention**.
Dans ce mécanisme nous avons en entrée : $\{x_1, \dots, x_n\}$ une séquence de plongement

i. Projections

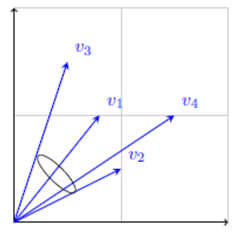
$$k_i = (XK)_i$$
$$q_i = (XQ)_i$$
$$v_i = (XV)_i$$

ii. Calcul des poids d'attention

$$a_{ij} = \frac{\exp(q_i^t k_j)}{\sum_{j=1}^{n} \exp(q_i^t k_j)}$$

iii. Plongements finaux

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



Les plongements finaux vivent dans le **cône** porté par les projections en valeurs, comme illustré ci-contre.

3. Les tâches de classification

Nous décidons de travailler sur les tâches de classification suivantes :

- **E-SNLI** [Camburu et al, 2018] : tâche d'inférence en langue naturelle
- **HateXplain** [Mathew et al, 2021] : détection du discours haineux
- **YelpHat** [Sent et al, 2020] : tâche de classification en polarité

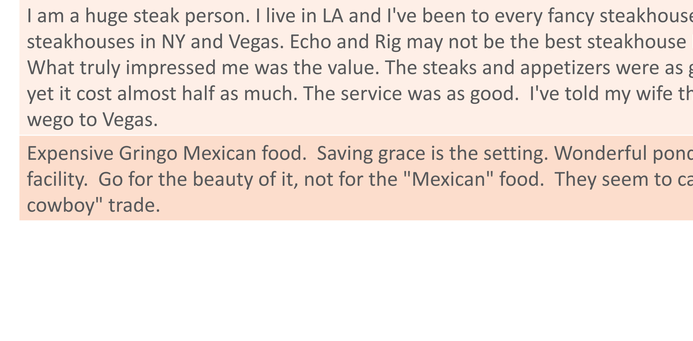
4. Un cône pour une classe ?

Les plongements s'organisent dans des cônes (résultat des expérience passées). Une question naturelle : **ce cône caractérise-t-il la classe ?**
Nous travaillons ici avec **BERT** [Devlin et al, 2019].

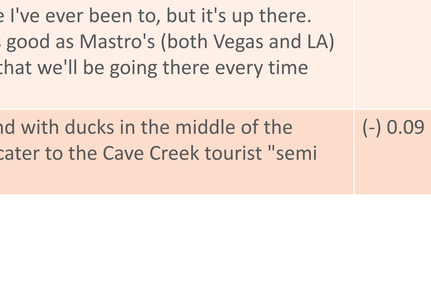
Expérience des représentants

- Pour chaque classe (c) on définit un représentant r(c) de la classe. Un représentant étant le plongement du *token* [CLS] d'une phrase bien classée (tirée aléatoirement).
- Pour les phrases restantes nous effectuons : $\cos(r(c), [CLS])$ puis nous distinguons chaque phrase par sa classe d'appartenance.

Résultats



(a) E-SNLI



(a) YelpHat

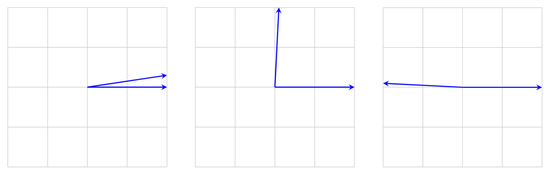
Exemples sur YelpHat

Phrase	(class) 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. Echo 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. I've told my wife that we'll be going there every time wego 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

5. Construction de l'anisotropie

Construction d'un modèle basé simplement sur le mécanisme d'auto-attention : **affranchissement des poids pré-entrainés**

Métrique de similarité : $sim(A) = \frac{2}{n(n-1)} \sum_{i<j} (AA^t) . \frac{1}{\sqrt{diag(AA^t) diag(AA^t)^t}}$



(a) $SIM(A) \approx 1$ (b) $SIM(A) \approx 0$ (c) $SIM(A) \approx -1$

Nous mesurons la valeur de cette métrique aux différents étages de l'architecture au niveau des **clés (K)** et des **valeurs (V)**

Résultats des mesures de similarités

		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
A=K	L=1	0.713	-	-	-	-	0.578	-	-	-	-	0.657	-
	L=2	0.691	0.683	-	-	-	0.649	0.556	-	-	-	0.719	0.489
	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	-	-
A=V	L=1	0.542	-	-	-	-	0.372	-	-	-	-	0.524	-
	L=2	0.510	0.740	-	-	-	0.409	0.511	-	-	-	0.182	0.746
	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	-	-

Lien avec la distribution des poids d'attention

- Utilisation des poids d'attention pour expliquer les décisions à postériori.
- Les poids d'attention déterminent la direction des plongements.
- Cependant les poids d'attention sont trop uniformes : tentatives pour les rendre parcimonieux

Régularisation par l'entropie [Nguyen et al, 2022] (1 tête 1 couche) : $\mathcal{L}(s) = \mathcal{L}(s) + \boldsymbol{\lambda} H(a_{0j})$

		$\lambda = 0$	$\lambda = 0.0001$	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$
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**	**0.484**

6. Conclusions

- En classification les plongements s'organisent des cônes**, la direction du cône dépendant de la classe et non de la phrase.**
- Cette organisation en cône se construit à l'intérieur du modèle : au niveau des clés et des valeurs.
- Si on vient perturber la distribution des poids d'attention **le modèle compense** pour ne pas changer la direction du cône.

GitHub

Contacts

loic.fosse@insa-rennes.fr / loic.fosse@orange.com
guig@irisa.fr
duc-hau.nguyen@irisa.fr
pascale.sebillot@irisa.fr

Ce travail a été réalisé dans le cadre d'un projet d'initiation à la recherche en Master 2. Loïc Fosse est actuellement en stage à Orange Labs.

Bibliographie / sources

- Camburu, O. M., Rocktäschel, T., Lukasiewicz, T., & Blunsom, P. (2018). e-snli: Natural language inference with natural language explanations.
- Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.
- Fosse, L., Nguyen, D. H., Sébillot, P., & Gravier, G. (2022, June). Une étude statistique des plongements dans les modèles transformers pour le français.
- Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2021, May). Hatexplain: A benchmark dataset for explainable hate speech detection.
- Nguyen, D. H., Gravier, G., & Sébillot, P. (2022, June). Filtrage et régularisation pour améliorer la plausibilité des poids d'attention dans la tâche d'inférence en langue naturelle.
- Sen, C., Hartvigsen, T., Yin, B., Kong, X., & Rundensteiner, E. (2020, July). Human attention maps for text classification: Do humans and neural networks focus on the same words?