Séparer la sémantique de la position dans les transformers: Approche

L'idée est de réserver certaines dimensions de l'embedding pour être positionnelles, exclusivement. C'est un peu comme l'embedding absolu, mais au lieu d'ajouter la représentation positionnelle à celle sémantique, on lui dédie des dimensions!

$$d_{\text{model}} = d_{\text{pos}} + d_{\text{sem}},$$

L'embedding de chaque token peut donc être écrit comme $\mathbf{e}_i = \left[\mathbf{p}_i \parallel \mathbf{s}_i \right] \in \mathbb{R}^{d_{\text{model}}}$, avec $\mathbf{p}_i \in \mathbb{R}^{d_{\text{pos}}}$ (partie positionnelles du token) et $\mathbf{s}_i \in \mathbb{R}^{d_{\text{sem}}}$ (partie sémantique du token).

La taille de p_i (nombre de dimensions positionnelles) est un hyperparamètre. On a d'abord testé $d_{\text{pos}} = d_{\text{head}} = \frac{d_{\text{model}}}{\text{nb\_head}}$.

Afin que le sémantique n'influe pas sur les dimensions positionnelles, nous devons prendre quelques précautions, notamment modifier W_V, W_O ainsi que le MLP.

Au niveau de l'attention

Nous devons forcer W_V à être diagonale blocks:

$$W_V = \begin{pmatrix} W_{\text{pos}} & 0 \\ 0 & W_{\text{sem}} \end{pmatrix}$$

Pour avoir:

$$V = \left[\mathbf{p} \parallel \mathbf{s} \right] \times \begin{pmatrix} W_{\text{pos}} & 0 \\ 0 & W_{\text{sem}} \end{pmatrix} = \left[\mathbf{p}W_{\text{pos}} \parallel \mathbf{s}W_{\text{sem}} \right]$$

Ainsi, en multipliant par l'attention:

$$A = \text{softmax}(\frac{\mathbf{Q} \cdot \mathbf{K}}{\sqrt{d_{\text{head}}}}) \in \mathbb{R}^{n \cdot n}$$

On a: $pW_{\text{pos}} \in \mathbb{R}^{(n \times d_p)}$ $sW_{\text{sem}} \in \mathbb{R}^{(n \times d_s)}$

Et plus précisément, la somme dans: $z_{i,k} = \sum_{j=1}^{d_{\text{model}}} A_{i,j} V_{i,k}$ se passe sur la ligne j et la colonne k fixes. Ainsi, d_{pos} et d_{sem} ne se mélangent pas.

Ensuite, pour le MHA (multi-head), on concatène la sortie de toutes les têtes:

$$\text{MHA} = \left[\mathbf{A}^{h_0} V^{h_0} \parallel \mathbf{A}^{h_1} V^{h_1} \parallel \cdots \parallel \mathbf{A}^{h_H} V^{h_H} \right] \in \mathbb{R}^{n \times (H \times d_{\text{head}})}$$

Au niveau de l'output

Comme pour la Value, on doit s'assurer que cette transformation linéaire ne mélange pas le sémantique et positionnel. On fixe donc W_O diagonale blocks, également:

$$W_O = \begin{pmatrix} W_{O_{\text{pos}}} & 0 \\ 0 & W_{O_{\text{sem}}} \end{pmatrix}$$

Ainsi:

$$\text{attn\_out} = \left[\mathbf{A}^{h_0} V^{h_0} \parallel \mathbf{A}^{h_1} V^{h_1} \parallel \cdots \parallel \mathbf{A}^{h_H} V^{h_H} \right] W_O \in \mathbb{R}^{(n \times d_{\text{model}})}$$

Au niveau du MLP: un peu plus compliqué

Nous devons réaliser 2 MLP différents pour s'assurer ici que les dimensions ne se mélangent pas: un MLP sur les dimensions positionnelles, et un autre sur les dimensions sémantiques. Le résultat final sera tout simplement une concaténation des 2.
On applique d'abord une projection linéaire pour obtenir le mlp_pre. À cette étape, on transforme la sortie de l'attention (AttnOutput) vers une taille intermédiaire, généralement $4 \times d_{\text{model}}$. Il est donc important de veiller à ne pas recombiner les dimensions positionnelles et sémantiques :

$$\text{mlp\_pre} = [\text{AttnOutputPos}\, W_I^{\text{pos}}; \text{AttnOutputSem}\, W_I^{\text{sem}}]$$

Avec :

$$\begin{cases} \text{AttnOutputPos} = \text{AttnOutput}[\cdot, : \text{pos}] \\ \text{AttnOutputSem} = \text{AttnOutput}[\cdot, \text{pos} :] \\ W_I^{\text{pos}} \in \mathbb{R}^{d_{\text{pos}} \times (4 \times d_{\text{pos}})} \\ W_I^{\text{sem}} \in \mathbb{R}^{d_{\text{sem}} \times (4 \times d_{\text{sem}})} \end{cases}$$

Ensuite, on applique les fonctions d'activation (séparemment sur la partie positionnelle et sémantique), donc on obtient:

$$\text{mlp\_post} = [\text{ACT}(\text{AttnOutputPos} W_I^{\text{pos}}); \text{ACT}(\text{AttnOutputSem} W_I^{\text{sem}})]$$

$$\text{mlp\_out} = [\text{mlp\_post\_pos} W_P^{\text{pos}}; \text{mlp\_post\_sem} W_P^{\text{sem}}]$$